Modeling, scaling and sequencing writing phases of Swiss television journalists

Supervisors
Prof. em. Dr. Iwar Werlen
Prof. Dr. Daniel Perrin

Affiliation
Doctoral thesis
Institute of Linguistics
Faculty of Humanities
University of Bern

Submission date
June 30, 2017

Author
Mathias Fürer
Zinistrasse 8
8004 Zürich
+4176417 71 93
mathias.fuerer@students.unibe.ch

Table of contents

Acknowledgements

First, I express my gratitude to my supervisors, Prof. em. Dr. Iwar Werlen and Prof. Dr. Daniel Perrin for their institutional support, their patience and their insightful comments on my work. And thanks, Daniel, for providing me time to focus. My great thanks go as well to all the journalists who patiently collaborated in the course of the research. Special thanks go to Thomas Gantenbein and Beate Sick for their support in statistical matters, to the Order of Saint Benedict in Disentis for providing me a calm room to work, to my family and all my friends, especially Kanan, for their curiosity and their patience.

1. Introduction

Writing is of decisive importance for a civilized society. Without writing our cognitive processes are limited to our attention span and our memory. Further, without writing we would not possess a widely understandable means to transmit complex information through space and time. It is unsurprising that in what may be the first Hollywood blockbuster to star a linguist as protagonist – *Arrival* –, the communication between human and alien was not achieved through spoken, but through written language.[1] Written language allows for time to analyze and does not urge immediate response on the verbal, paraverbal, and the nonverbal level of communication as spoken language does.

The product of writing, the final text, is the principal object of study for a number of disciplines in the humanities, including, among others, philosophy, history, and literary studies. Through researching texts, we can identify many premises of thinking, histories of ideas can be reconstructed, and language use can be described and analyzed. To a certain degree it is also possible to use the writing product to infer the writing process, especially if provided access to preliminary versions of the text or aids such as marginal notes (Grésillon & Lebrave, 2008). Unfortunately, however, the motives for formulating a sentence this way or that way at a certain position in the text can only be guessed – the justifications for most revisions remain obscure.

It is exactly this question – What do people think when they write? – that drives my overall investigations into writing processes in general, and writing phases in particular. I want to find out what kind of strategies they adopt and assess how these work out. The process of writing, here understood as the linearization and materialization of human's basically associative thinking, reveals a lot about how we picture the world and how we relate to it. The same goes for professional writers, except in their cases, not only are their personal beliefs revealed, but their professional standards as well.

My principal interest in the writing of journalists stems from the fact that journalists construct images of reality that matter for society.[2] Although today's digitalized world provides the technical means of mass communication for virtually everyone, other sources of reality construction have an increasing impact on what is perceived as relevant. Yet the journalists' role as selector, distributor, and magnifier of stories remain unmatched.[3]

Consequently, the investigation of journalistic writing processes exhibits a rewarding research endeavor, if not a challenging one. Whereas the empirical research of writing processes in educational settings can be traced back to the to the 19th century (Knobloch, 2000), the number of researchers investigating text production[4] processes in professional settings, and who actually log writing processes in some way, can still be counted on two hands (Ehrensberger-Dow & Perrin, 2015; Mariëlle Leijten, Van Waes, Schriver, & Hayes, 2014; Perrin, 2013; Schrijver, Vaerenbergh, Leijten, & Van Waes, 2014).

---

[1] Despite the simplification, dramatization, and exaggerated and outdated portrayal of the Sapir-Whorf-Hypothesis, I recommend watching the movie for its illustrative application of Chomsky's (1995) minimalist program for linguistic theory.

[2] In terms of reality perception and construction, I follow Gabriel's (2015) *new realism* that adds the real existence of certain objects to the radical perspectivism of constructivism.

[3] This is as well reflected by the fact that, in terms of reach and impact, most enterprises still favor media coverage over coverage in social media. This favoring is not only because they can reach their audience over social media anyway if they invest sufficiently, but also because the (planned) interaction effects between media coverage and coverage in the social media multiply the effect on their audience.

[4] The term "text production processes" accounts for the fact that the writing of a text often assumes several other preceded or parallel activities than the writing itself. In the case of television newswriting, desk research, phone calls to planned protagonists of the news item, and editorial meetings are examples of such activities.

Unsurprisingly, given that writing is a basic skill that is taught all over the world, most publications dealing with writing processes originate from the educational context. Consequently, researchers who investigate and strive to improve the teaching of writing to children and juveniles are more numerous than those who want to achieve the same for adult professionals. Besides, in the latter group the majority investigate the writing processes of academics. Based on my literature review and attendance at numerous conferences[5] over the past five years, it has become clear that empirical studies of writing processes at the workplace are sparse.

More systematic assessments also come to the same result. In a meta-analysis of context in writing process research, Glopper, Kruiningen, and Hemmen (2014) drew on two separate samples, one for writing process studies in educational settings, another for professional settings. While they achieved their desired sampling of 45 studies for the educational domain, they only managed to find 25 studies for the professional domain, concluding that "The small sample size reflects the scarcity of writing process research in this domain" (p. 20),. All this is not necessary to mention, that only very few studies in this small sample drew on logged writing processes data.

The scarcity of empirical studies drawing on logged writing process data has consequences for the investigation of my even more specific object of research: writing phases. In light of the fact that the writing process as whole is very complex and consists of various activities, it corresponds to the sciences' ideal of differentiation to divide the writing process into several phases.[6] Although models of writing exist that include phase concepts, they tend to be empirically vague, as Perrin, Fürer, Gantenbein, Sick, and Wildi (2011) put it:

> The complex interplay of writing and reading phases that constitutes text production processes has not yet been tracked in natural settings with large scale samples to gain the data corpora needed for adequate theories of writing that are empirically validated. Therefore, the theories, models and approaches presented so far have tended to be empirically vague: they are speculative, based on experiments or on single case studies. As a consequence, good practice models of writing lack empirically testable explanations of writing processes in general and phases in particular. We consider this to be a deficiency for systematic teaching and evaluation of writing. (p. 2)

Thus, in this work I am traversing a wealth of unexplored territory. In order to demarcate my research, I start chapter two by situating it within the relevant disciplines: linguistics, applied linguistics, mass communication research, and media linguistics. Second, I differentiate my phase concept from previous ones and present the writing process model of situated newswriting of Perrin (2013) within which I situate my research. Third, I introduce the Dynamic Systems Theory and substantiate why it is well suited to explain the often non-linear dynamics of writing.

In the third chapter of the book, I lay out the qualitative and quantitative methods I use to analyze one of the most extensive data collections of writing processes in natural settings ever collected. As undergraduate student, I contributed to ethnographically collecting 120 writing processes from 15 Swiss television journalists under the framework of the *Idée Suisse* project (see section 3.1.1). As doctoral student, I transformed, annotated, and coded these 120 writing processes as a part of the *Modeling Writing Phases* project (see section 3.2.2) – a total of 16'847 revisions. In section 3.1, the building and transforming of this corpus is described, and in the subsequent section I explain how the qualitatively

---

[5] For an overview of the conferences at which I have presented, see section 7.5 in the appendix.

[6] I prefer the term "phase" because it entails a temporal aspect and is not yet occupied by other similar, but different concepts in writing process research, such as "episode."

identified writing phases are modeled by machine learning methods.[7] Chapter three ends with remarks about how the methods were combined.

In chapter four I present the results of the analysis in the form of four levels of writing phases that scale from few revisions to bigger text parts. On the largest level of the identified writing phases, the effects of sequence of the lower level are presented. In addition, some general properties and the validity of the modeling process are discussed.

In the fifth and last chapter, the results of the study are interpreted and their significance for the research of writing processes in general, and writing phases in particular, are discussed.

[7] This is the first time that machine learning methods were applied on writing process data. In other linguistic fields, they have already been successfully applied, especially in text linguistics: The computer scientists Potthast, Köpsel, Stein, and Hagen (2016) use a random forests model (see section 3.2.2) to detect click-bait tweets.

2. Theory

In this chapter, I present the epistemic milestones that preceded and allowed for the chosen focus on of this dissertation: writing phases. The investigation of writing phases in journalistic contexts requires knowledge from within several disciplines. Linguistics and applied linguistics contribute methods for analyzing language, which are used in every step of news production – from the generation of the first idea to the broadcasted news item. Media and communication studies provide relevant background information about the configuration of the media system, the socioeconomic status of journalists, and media content. Finally, media linguistics apply the methods of linguistics on the language of the media

2.1. Disciplines involved

Researching writing processes is a multidisciplinary endeavor. Several disciplines, such as linguistics, applied linguistics, psychology, educational sciences, sociology of work, literary studies, and others contribute to this field.

In terms of disciplines, my starting point is linguistics, the scientific study of natural language form, meaning, and use – whether spoken, written, or signed (Bussmann, 2008; Perrin, 2013, p. 16). Writing in professional settings such as television newsrooms, is a specific form of language use. As such, it is mainly investigated within applied linguistics, a branch of linguistics that stresses the focus on language use. However, De Saussure (1916), often portrayed as the founder of modern linguistics and without a doubt one of the most influential linguists of the twentieth century postulated decisively that language use – in all its forms – is a research field of linguistics:

"La matière de la linguistique est constituée d'abord par toutes les manifestations du langage humain, qu'il s'agisse des peuples sauvages ou des nations civilisée, des époques archaïque, classiques ou de décadence, en tenant compte, dans chaque période, non seulement du langage correct et du 'beau langage', mais de toutes les formes d'expression" (p. 20).[8]

Hence, the reasons for the constitution of applied linguistics lie less in the subject of study than in the aimed effect of the research. Applied linguistics connects "knowledge about language to decision-making in the real world. Generally speaking, the role of applied linguists is to make insights drawn from areas of language study relevant to such decision-making" (Simpson, 2011, p. 1).

Historically, applied linguistics emerged in the 1980s to academically underpin the study of language and teaching.[9] Nowadays, although the bulk of research conducted under the label of 'applied linguistics' still deals with practical problems of language acquisition and teaching, it has evolved into a discipline for "the theoretical and empirical investigation of real-world problems in which language is a central issue" (Brumfit, 1995, p. 27). Furthermore, the separation between general linguistics and applied linguistics is a topic of its own discussion (Widdowson, 2000) and can be related, among other things, to the seminal influence of Chomsky (Cook, 2003, p. 9) who conceives language in his generative linguistics as essentially biological and not cultural or social (Chomsky, 2016).

---

[8] My translation: All manifestations of human language are subjects to linguistics, whether they originate from savage people or from civilized nations, from archaic, classic or decadent eras. For all periods not only the correct language or the 'right' language but all forms of expression have to be taken into account.

[9] Another important target of intended impact by applied linguists is language planning and language policing. In this field some applied linguists favor for constructed or planned languages what can be interpreted as violation of general linguistics limitation to natural language – depending on the interpretation of natural.

Since I investigate writing phases in the domain of journalism, another bundle of disciplines is relevant for my object of study: media studies, mass communication research, and communication studies.[10] These three disciplines provide knowledge about the economic and political properties of media systems and the socioeconomic status of journalists, offer theories of the effects of mass communication and of audience uptake, and investigate media content and the development of media technology. In the social science strand of media studies, surveys and content analysis are widely used methods, whereas in the cultural studies strand, hermeneutic approaches are adopted. But neither media studies nor its subdisciplines consider a systematic consideration of what media content is made of: language.

To fill this gap, linguists who are interested in the language of the media developed and fostered the interdisciplinary field of media linguistics (Bell, 1991; Luginbühl & Perrin, 2011; Perrin, 2015). As media linguistics investigates "language use in public discourse and the media" (Perrin, 2013, p. 16) and journalistic text production is situated in this field, my research focus – writing phases of television journalists – fits best within media linguistics.

2.2. Writing phases as temporal and functional units

There is no consensual definition of a writing phase in the scholarly discourse. Further, it takes a considerable amount of interpretation to identify phase concepts in studies that are comparable to mine. But no scholar of writing research would contest the importance of the oft-cited theory of writing processes that was published 1981 as a result of an interdisciplinary collaboration: the *cognitive process theory* by the Anglicist, Linda S. Flower and Psychologist John R. Hayes, both working at the Carnegie Mellon University at the time.

Flower and Hayes (1981) apply insights from cognitive psychology to the field of writing research and thus elaborate the stage models by Britton, Burgess, Martin, McLeod, and Rosen (1975) and Rohman (1965). They propose a three-stage conceptualization in the writing process part of their model[11], consisting of planning, translating and reviewing (see Fig. 1). During the planning process "writers form an internal representation of the knowledge that will be used in writing" (Flower & Hayes, 1981, p. 372). Translating refers to the process of putting ideas into visible language and reviewing includes evaluating and revising the text produced so far. Flower and Hayes emphasize "that people do not march through these processes in a simple 1, 2, 3 order" (Flower & Hayes, 1981, p. 375). Instead, these processes may occur at any time of the writing process although they are not equally probable at all time.

[10] Despite the terms have different etymologies and signify different research traditions they are not used consistently. However, all three disciplines involve communication mediated through mass media.

[11] Other parts of their model were the writer's long-term memory and the task environment (Flower & Hayes, 1981, p. 370).

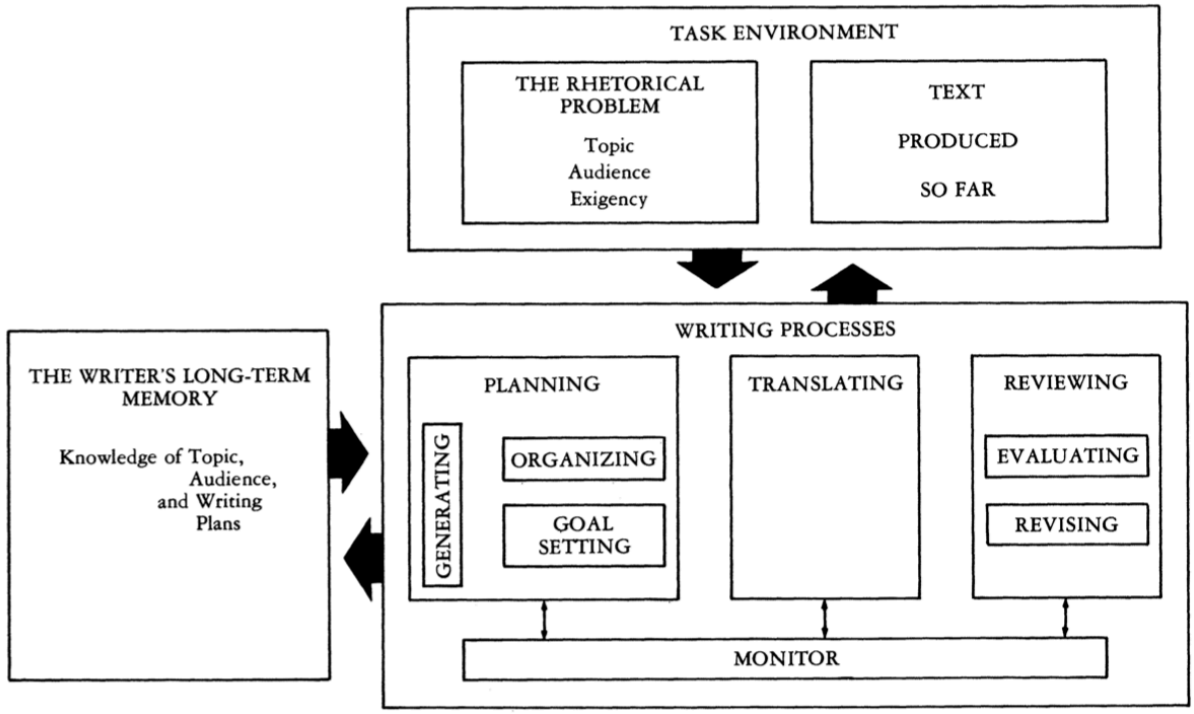

Fig. 1: Flowers and Hayes' (1981) cognitive writing process model

Flower and Hayes' (1981) model can be misunderstood as linear due to their suggested flow chart. To avoid misunderstanding, they decisively assure its non-linearity:

What the arrows *do not mean* is that such information flows in a predictable left to right circuit, from one box to another as if the diagram were a one-way flow chart. This distinction is crucial because such a flow chart implies the very kind of stage model against which we wish to argue. One of the central premises of the cognitive process theory presented here is that writers are constantly, instant by instant, orchestrating a battery of cognitive processes as they integrate planning, remembering, writing, and rereading. The multiple arrows, which are conventions in diagramming this sort of model, are unfortunately only weak indications of the complex and active organization of thinking processes which our work attempts to model. (p. 387)

With this quote Flower and Hayes make clear that they do not conceptualize writing phases as temporally delimited units. For them, while writing phases may be temporally divided, they are very brief and occur associatively and seemingly chaotically. In the end, Hayes – also in his recent model (Hayes, 2012) – and other scholars who investigate writing processes from the perspective of cognitive psychology (e.g. Alamargot & Chanquoy, 2011; Grabowski, 1996; Kellogg, 1996; Rijlaarsdam & Van den Bergh, 2006) offer no phase concept that includes a verifiable time dimension. After all, they are interested in which resources of the working memory are used for what basic processes of writing – in the case of T. Kellogg, P. Whiteford, E. Turner, Cahill, and Mertens (2013) the visual-spatial sketchpad, the central executive and the phonological loop) the visual-spatial sketchpad, the central executive, and the phonological loop. For these authors, those resources are planning, translating, programming,[12],

[12] With *programming,* the authors refer to the programming of the motor that is responsible for the desired motor implementation of the writing. In particular, they are interested in how much cognitive load the motor programming occupies at the expense of other cognitive processes. For children, the automation of this lower level processes is important for having more resources of the central executive available for "the higher order demands of planning ideas, generating text, and reviewing the work produced thus far" (T. Kellogg et al., 2013, p. 163).

executing, reading and editing. Hayes himself updated his and Flowers model from 1981 (see Fig. 2) but has not incorporated a temporal dimension for writing phases comparable to mine,.

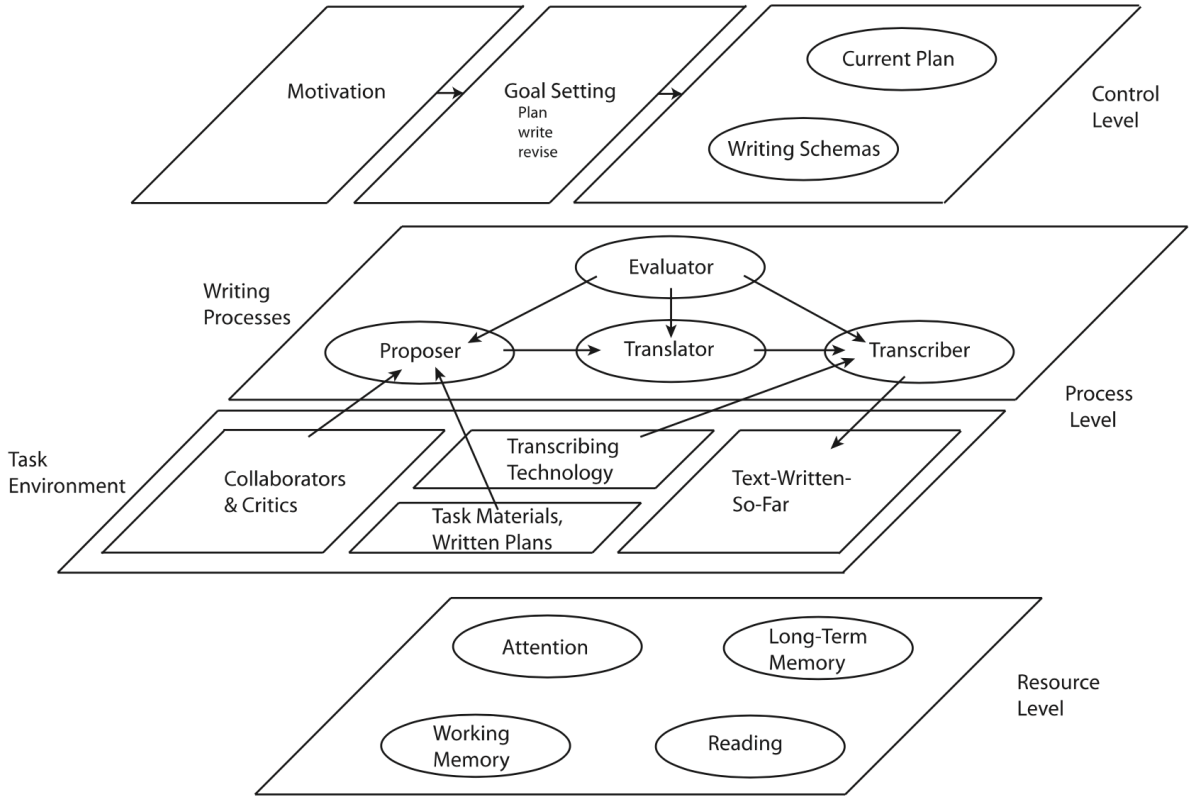

Fig. 2: Hayes' (2012) model of writing processes

Due to the inclusion of a temporal and functional dimension for the definition of writing phases, my concept of writing phases differs substantially from the cognitive psychology paradigm. The same is true for the framework of activity theory (Bazerman, 2003; Bracewell, 2003; Russell, 1997) which investigates "material or symbolic means that people use to accomplish objectives that carry the effect of these factors and serve to mediate the setting of an objective and its achievement" (Bracewell, 2003, p. 513).

Another paradigm – created not primarily to investigate, but to conceive writing differently – originates from Elbow (1998b)[13] who criticizes outlining before writing:

> This idea of writing is backwards. That's why it causes so much trouble. Instead of a two-step transaction of meaning- into-language, think of writing as an organic, developmental process in which you start writing at the very beginning – before you know your meaning at all – and encourage your words gradually to change and evolve. Only at the end will you know what you want to say or the words you want to say it with. (p. 15)

In his book *Writing without teachers* Elbow promoted epistemic writing, i.e. writing as a mean to generate ideas.[14] The underlying principle of Elbow's approach is that writing competence has to be built out of experiences, and experiences are inherently subjective. Consequently, every writer starts

[13] This book, *Writing without teachers*, was first published in 1973.

[14] In the US-American scholarly discourse the term "epistemic writing" is not very common. They prefer the term "reflective writing", defined as "a dialectical process by which higher-order knowledge is created through the effort to reconcile lower-order elements of knowledge" (Scardamalia, Bereiter, & Steinbach, 1984, p. 173).

from a different starting point and then follows different paths. Elbow identifies two stages in a writing process – creative writing and critical revising - and a lot of his effort goes into convincing his readers that they should separate the two: "If you separate the writing process into two stages, you can exploit these opposing muscles one at a time: first be loose and accepting as you do fast early writing; then be critically tough minded as you revise what you have produced" (Elbow, 1998a, p. 9).

However, Elbow's two stages of writing do not offer a concept of temporally delimited writing phases comparable to mine. His ideas are fed by experience in training of writing and have normative character. They may help a writer to discern the two stages of creative writing and critical revising, and by doing so, to avoid a writer's block provoked by early revising. But as analytical units of a writing process they are fairly rough: It is desirable to know more about what happens within Elbow's two stages if a writer sticks to his concept. And what about creative ideas that evolve from critical revising? More fine-grained and less normative analytical units grasp more of the complexity of a writing process.

I identify two main reasons why my concept of writing phases – defined as empirically observable temporal segments of writing processes that are dominated by a particular writing activity – has only a few followers. The first reason is because it is seen as too complex to identify all of the individual decisions that are involved in writing. Writing – in its non-collaborative parts – is a highly independent, often isolated activity that is realized mostly by cognitive processes which are difficult to access via traditional research methods (Wrobel, 2000, p. 458). Moreover, results from introspection, think-aloud, and retrospective protocols (see section 3.1 for an explanation of the protocols) hint at the complex interplay of the hierarchically organized processes involved in writing. For example, a writer may change the goal of a whole text because she or he had an idea while writing.[15]

The second – and more important – reason for not conceptualizing writing phases as the verifiable temporal segments of writing processes I identify, is the lack of general empirical analyses of writing processes. Numerous researchers theorize about writing processes without verifiably founding their conclusions on empirical data. Others proceed empirically, but only investigate a limited aspect, e.g. planning, drafting, formulating, or revising. Keeping the legal and technical barriers for recording writing processes in natural settings in mind (see section 3.1.1), it is understandable that more studies with my perspective on writing phases have not been employed, especially given the fact that doing so requires a considerable amount of data of the writing process.

One model that has been built on empirical evidence in natural settings is the helix of situated newswriting. Perrin (2013, p. 151) differentiates 16 activity fields of newswriting (see Fig. 3) that are integrated in the helix (see Fig. 4). This writing process model allows for temporally and functionally delimited writing phases. In the case of newswriting, Perrin developed a fine-grained categorization system with hundreds of subcategories for these 16 activity fields via the abductive coding of thousands

[15] An illustrative case analysis comes from Perrin (2012a): A television journalist writes a news item about demonstrations in Lebanon. While writing about how the people traveled to the demonstration, he changes the word *express* (fast) into *tranquille* (calm) to describe their way of passage. Out of habit the journalist initially used the standard expression *voie express* but then changed it because it did not correspond to the calm pictures he planned to use that show peaceful demonstrators on boats. In addition, he then realized that these pictures combined with his adjusted adjective for once give him the opportunity to portray these events in the Middle East against the Western bias of violence and conflict. Having spent several years in the middle east as a correspondent, he seized the opportunity and chose "calm" as a leitmotif for his news item.

of revisions.[16] Thus, he links single and groups of revisions to structure, functions, and environments of writing.

Structure	Function	Environment
Reading sources	Finding the sources	Handling social environment
Reading own text	Limiting the topic	Handling tools environment
Goal setting	Taking own positions	Handling task environment
Planning	Staging the story	Comprehending the task
Controlling	Establishing relevance	Implementing the product
Monitoring		

Fig. 3: Perrin's (2013) activity fields of newswriting

Whereas the structure and the environment of writing can be found as well in Flowers and Hayes' model (1981), Perrin (2013) adds functions: *finding the sources*, *limiting the topic*, *taking own positions*, *staging the story* and *establishing relevance for the audience*. Furthermore, he visually highlights the importance of reading during the whole writing process by integrating the *reading of source text* and the *reading of own text*.

[16] In interactive knowledge map of these categories can be accessed in Perrin (2011).

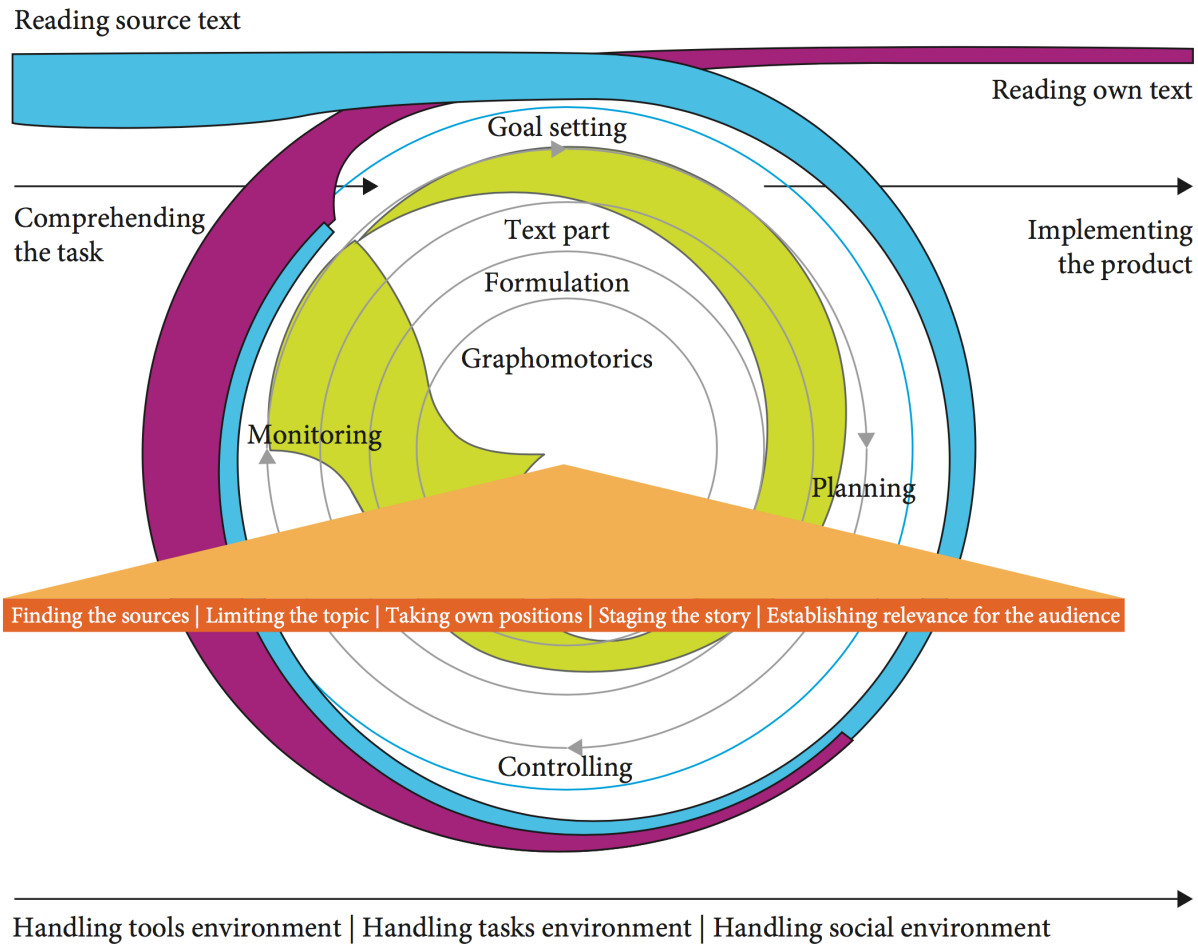

Fig. 4: Perrin's (2013) model of situated newswriting

Perrin (2013) developed the model of situated newswriting out of a more general theory in which he combines the applied linguistics perspective with the writing research perspective: "The theoretical perspective … opens up a space of systematically conceivable options, reaching far beyond the range of the experienced or identified practices in newswriting and research into it…. In contrast, the practical perspective … focuses on the reality of daily newswriting. It helps to explain in detail what happens, matters, and works under which conditions" (p. 152).

Despite these carefully set limitations, I consider the more specific model of situated newswriting above as fruitfully applicable to more domains than newswriting. Therefore, I use the word "writer" to indicate that the point also fits with a more general conception of writing. If I mean to restrict my comments to the domain of journalism, I use "journalist".

2.3. Grasping the dynamics of writing phases

The dynamics of writing and writing phases are complex. The sequentiality and the postulated and in section 4 described scalability of writing phases call for a suitable theoretical framework and appropriate terminology. The *Dynamic Systems Theory* offers both, because it "embraces complexity, interconnectedness, and dynamism, and makes change central to theory and method" (Larsen-Freeman

& Cameron, 2008, p. 1).[17] I am following the applied linguists Larsen-Freeman and Cameron (2008) who define a dynamic system as a "system with different types of elements, usually in large numbers, which connect and interact in different and changing ways" (p. 26).

The idea behind Dynamic Systems Theory has its roots in mathematics, physics, and biology, and is used to describe autopoietic systems, i.e. systems that continually change and build new structures while maintaining their identity.[18] That is exactly what writers in general, and journalists in particular, do: They produce new texts by adapting their personal writing strategies to a constantly and often non-linearly changing environment – all while maintaining their identity. This is particularly true for journalists since their environment is more complex and more susceptible to change than the environment of a lonely writer or a fourth grader.

The key features of complex systems in terms of Dynamic Systems Theory are:

a) heterogeneity of elements or agents

b) dynamics

c) non-linearity

d) openness

e) adaptation

a) In the case of television journalists, the heterogeneity of elements means that they have to combine the three modes of video communication (text, images and sound) in a way that clearly gets their message across. The heterogeneity of agents can be observed in editorial meetings when more than ten journalists, each with their own unique approach to the news, gather to discuss the topics of the day, or when the different roles in the newsroom (producer, speaker, editor-in-chief, and other journalists) are considered together. The heterogeneity of agents can as well be observed externally: journalists speak to artists, lawyers, farmers, CEOs and laypersons in order to produce their item.

b) The dynamics in the system of daily news production business are lively and seldom linear: New information may render a carefully written news story 'yesterday's within' seconds. To accommodate such a c) rapidly changing environment, successful television news journalists remain d) open for the unexpected and find emergent solutions to cope with the unpredictability of reality, while simultaneously quickly and decisively e) adapting their strategies to changes.

Besides the key features of dynamic systems presented above, Dynamic Systems Theory also offers an intricate, but effective terminology to describe change generally. In the following paragraphs, I will introduce a simplified version of Dynamic Systems Theory, which will be used throughout the remainder of the book to assist in describing the dynamics of writing phases.

In terms of Dynamic Systems Theory, complex systems such as television newsrooms, are situated in a "state space" (Larsen-Freeman & Cameron, 2008, p. 46). The state space of a system includes all of its possible states, no matter if they are realized or not. Continuing with the example of the television newsroom, editorial meetings or the collaborative video editing with a video editor may be interpreted

---

[17] The idea of complex systems and dynamics of change carries many other names: *Complex Adaptive Systems Theory*, *Complexity Theory*, *Complex Systems Theory*, *Dynamical Systems Theory*, *Chaos Theory* and others. I follow Perrin (2013) who chooses the term *Dynamic Systems Theory* because of its focus on dynamics and thus processes.

[18] See Larsen-Freeman and Cameron (2008, pp. 2–4) for a chronological account of the application of Dynamic Systems Theory.

as state spaces. Thinking of the countless options for writing, it becomes clear that the state space of writing has a very high number of possible states. If general writing is instead limited to writing for television news, then the number of state spaces is reduced. This can be, for example, due to the fact that the text has to be related to the two other modes, video and sound.

When a dynamic system changes it moves from one state to another and the path between those two states is called the "trajectory", in terms of Dynamic Systems Theory. This change is evoked by so called "attractors," which are preferred states of the system. If a writer changes from linear writing to revising, it can be interpreted as trajectory of the system from one state to another. Attractors can provide stability for the system, but can also bring about imbalance, or constrain its trajectory to other, more desirable states. For example, writing habits can facilitate writing, but they also constrain creativity.

Larsen-Freeman and Cameron (2008, pp. 56–57) report three types of attractors: "fixed point attractors", "cyclic attractors" and "chaotic attractors". A fixed point attractor represents a preferred and stable region in the state space for the system, such as the above mentioned states of writing and revising. They can also be, and are more suitably interpreted as, cyclic attractors, also called "closed loop attractors," which are two or more regions of the state space between those the system moves periodically. Finally, chaotic attractors exhibit regions of the state space where the trajectory of the system becomes unstable, "as even the smallest perturbation causes it to move from one state to another" (Larsen-Freeman & Cameron, 2008, p. 57). For example, the loss of the text produced so far due to technical problems can be interpreted as chaotic attractor.

Nevertheless, "chaotic" in this context should not be interpreted as random, but rather as less predictable and stable compared to the effects of the other two types of attractors. For the example of file loss this means that due to the systematic, and therefore normal time pressure in the daily news business, there is usually only very little time left to rewrite the news item after a file loss. This stressful situation may cause the journalist to quickly change from one text production activity to another. She or he may inform the producer, who is responsible for the whole edition of the news program, call a video editor for help, and start rewriting from memory. In terms of Dynamic Systems Theory, the system changes rapidly and unstably from one state to another.

Such chaotic attractors will sometimes result in "emergence", i.e. a phase shift to a region in the state space on a higher level where "the whole is more than the sum of its parts and cannot be explained reductively through the activity of the component parts" (Larsen-Freeman & Cameron, 2008, p. 59). Applied to the example of file loss mentioned above, the journalist may abandon the solution of rewriting the news item as it was before, in favor of writing a new item on the same topic, but with a different dramaturgy that is faster to implement because the journalist can reuse pieces from an already broadcasted item and combine them. This solution is even more emergent if by doing so, the message of the item shifts to a more favorable one, possibly one that the journalist already had, but abandoned due to lack of courage. Consequently, the final state space, i.e. the final news item, is more preferable than the one that the journalist initially aimed for.

There is an ongoing scholarly discussion concerning whether the Dynamic Systems Theory, and the approaches related to it, are something new (Larsen-Freeman & Cameron, 2008, pp. 6–9) and if they are "just" metaphors. Here I follow Larsen-Freeman and Cameron (2008) who, first frame metaphors not as "just," but as valuable scientific tools (pp. 11–15), and second, state that the usage of the metaphors of Dynamic Systems Theory eventually yield field specific theory, research, and practice (pp.

15–17). That is exactly what I will do in section 4 where I describe the dynamics of writing phases in terms provided by Dynamic Systems Theory.

Another reason why Dynamic Systems Theory exhibits a suitable framework for my research is because within it I can also situate the technique of iterative, recursive, and abductive coding that was applied to qualitatively identify writing phases. The success of this approach has been demonstrated by Michael Agar, a linguist, ethnographer, and anthropologist who present these three principles in his essay *We Have Met the Other and We're All Nonlinear: Ethnography as a Nonlinear Dynamic System* (M. Agar, 2004). He explains,

> An ethnographer engages in a cyclical process, modifying and trying out frameworks that initially didn't work until they work so well across so many kinds of data that they become candidates for an ethnographic conclusion. The processes of modification and validation are too complicated to detail here, but the critical point for now is that they typically require several cycles. They are *iterative*. And while in the middle of working on one rich point, another often comes up. What is an ethnographer to do? He/she now applies the same process in which he/she is currently engaged, only this time to a rich point that appeared as the process was ongoing. In other words, the cycle isn't only repeated over and over again. It is also applied within itself. The process is also *recursive*. (2004, p. 21)

Agar continues by relating the idea of "fractals" to his ethnographic work. For now, it suffices for us to grasp the two principles in the quote above that were applied to the qualitative coding: iterativity and recursivity. The third principle, abduction, also fits well into Dynamic Systems Theory because of its focus on what M. H. Agar (2010) calls "surprises" that are emergent solutions in terms of Dynamic Systems Theory. If we observe an emergent solution as a researcher, it will surprise us. Following abductive logic, we will have to look for the conditions that made this solution possible. Obviously, these conditions are as yet obscure to us, otherwise the emergent solution based on them would not have surprised us. Identifying these conditions means generating new knowledge.

2.4. Key terms

As I showed in the previous sections, writing research literature does not use a common terminology to describe what writing phases are. Accordingly, I begin by defining and differentiating the key terms I use for the analysis of writing phases such as "text production," "revision," and "writing phase."

2.4.1. Text production process and writing process

Text production process: An activity complex in which a written text is generated in order to accomplish a task.

Text production tasks combine mnemotechnical, epistemic, and communicative functions of writing, i.e. the authors write to develop thoughts and to save and share them. Tasks are set by the authors themselves or by others. An example from the *Idée Suisse* research corpus would be: to produce a two-minute media item on demonstrations in Lebanon, focusing on the demonstrators' perspectives.

Writing process: All of the activities involved in producing written language within a text production process. Writing combines material, cognitive, and social activities.

a) On a *material level*, writing processes can be observed as the situated activity of applying stretches of language onto an optically/sensually readable medium – or deleting them from it.

b) On a *cognitive level*, writing processes include all of the mental activities related to producing written language. They include thoughts that emerge from the interaction of the author's psycho-biography with the context, in particular with the sources and the text under construction.

c) On a *social level*, writing takes place within a context and alters this context as well, including its related social settings and collaborative practices. Key elements are cultural values, editorial norms, and organizational resources constraining or enabling decisions about text production.

The minimal analytical unit of the writing process is the revision, as represented in S-notation (2.4.2). The sequence of revisions in an entire writing process can be plotted in a progression graph (2.4.3.). At the same time, progression graphs facilitate the analysis of complex key segments of writing processes: the writing phases (2.4.4).

2.4.2. Revision and S-notation

In the present context of empirical writing research, revision refers to a micro-step in the process of writing and revising a text: an insertion or a deletion. An insertion is the process of adding a stretch of characters to an existing text. Pauses within an insertion do not delimit another revision, i.e. another insertion. A deletion is the process of eliminating any stretch of characters from a text.

Revision: The procedural micro-unit of writing processes that consists of a sequence of operations to either insert a single stretch of characters in a growing text or delete a single stretch of characters from it.

All text operations can be described in terms of insertions and deletions. When stretches of characters are overwritten, this is analyzed as deleting the old stretch and inserting a new one. Similarly, copy-pasting or moving stretches of characters in the text is analyzed as deleting a stretch at the old position and inserting the same stretch at the new position.

Not covered by the present definition of revision are more complex procedures, such as the overall process of revising a draft version of a text.[19] The result of such a process, i.e., the revision as a new version of the text, is not covered by the definition either. I consider these to be larger phases and products of writing processes.

Sequences of insertions and deletions can be described in S-notation. Wherever the writing is interrupted to delete or add something, S-notation inserts the break-character $|_n$ in the text. Deleted passages are enclosed in ^{n}[square brackets]^n and insertions in ^{n}{curly braces}^{n}, with the superscript numbers indicating the order of these steps.

S-notation: A transcription standard that marks insertions and deletions and indicates their sequence in the writing process.

In the following example from the *Idée Suisse* research corpus (see section 3.1) , the word *express* is deleted as revision number 20, before *tranquille* is inserted as revision 21. This does not happen until after the first version of this section of the text is written, as is evident from the deletion of the *e* further on in the text, which took place much earlier in the process, as revision 4. The underlining in the example indicates the text that appears in the final version (Ex. 1).

[19] Vandendaele, De Cuypere, and Van Praet (2015) systematize the meanings and usage of the term revision.

```
par la voie ²⁰[express]²⁰|₂₁²¹{tranquille}²¹ de la Médit⁴[e|₄]⁴érannée
```

Ex. 1: Exemplification of S-notation
Source: tsr_tj_070214_1245_guillet_libanon_snt_3

S-notation was invented by Py Kollberg and Kerstin Severinson Eklundh (Kollberg, 1997, 1998; Severinson-Eklundh & Kollberg, 1996a, 1996b). In the early days of computer linguistics, they developed the software *JEdit* for logging writing processes and *TraceIt* to transcribe the *JEdit*-logfiles into S-Notation (Nilsson & Kollberg, 1994). S-Notation depicts a very reduced way to represent writing processes in that it exclusively shows the ordinal sequence of insertions, movements, and deletions. Contemporary keystroke logging tools capture writing processes in much more detail: *Inputlog*, currently the most developed and established research tool for logging and analyzing writing processes, allows for measuring the transition between keys in milliseconds, logs mouse movements, as well as integrates speech recognition and eye tracking (M. Leijten & Van Waes, 2013; Van Waes & Leijten, 2005).

A potential critique of S-Notation is that it does not provide the accuracy and density of data accomplishable with contemporary keystroke logging tools. On the one hand, neuro- and psycholinguistic research requires dense and accurate data for some research questions, e.g. what effect dyslexia has on typing (Berninger, Nielsen, Abbott, Wijsman, & Raskind, 2008). On the other hand, this rich data can be difficult to interpret. The difficulty of interpretation already starts with the operationalization of a pause. If a researcher wants to log and analyze writing processes, then she or he has to decide what it means if there is no observable material writing activity and how long this inactivity has to last to constitute a pause. The scholars of writing processes who follow the paradigm of cognitive psychology (see section 2.2) often use a threshold of 1 to 2 seconds.[20] This exact value is not only arbitrary, but also tends to semiconsciously blind one to the fact that it is not yet possible – and will probably not be in the near future – to directly access the cognitive processes within those seconds. The pause threshold is an undoubtedly necessary assumption to investigate how "writing fluency and flow reveal traces of the underlying cognitive processes" (M. Leijten & Van Waes, 2013, p. 360). But stepping down from the interval to the ordinal scale, a step that S-Notation takes, means preserving informative data, but removing data points that are difficult to interpret.[21] The limitation of the ordinal sequence of revisions ensures the focus is on the bigger picture of the writing process and constitutes an adequate reduction of complexity for the analysis of a larger corpus of writing processes.

2.4.3. Progression graph

A progression graph is a figure showing all revisions occurring in a writing process as ordinal data. It relates the sequence of revisions in the writing process with the sequence of revisions in the text product. In doing so, it indicates how the writer moved with the cursor from revision to revision through the developing text. These cursor movements are interpreted as the writer's shifts in focus.

---

[20] An overview of the various thresholds can be found in Chenu, Pellegrino, Jisa, and Fayol (2014) and a discussion of the various functions of pauses in Wengelin et al. (2009).

[21] Inputlog allows also for revision analyses and output in S-Notation although this functionality is implemented only for text produced with the word processor software Microsoft Word (M. Leijten & Van Waes, n.d.).

Each data point represents one revision: the x-axis marks the progression in the process, the y-axis the progression in the product. This deviation of the standard coordinate system where the y-axis is drawn upwards has an illustrative purpose: from the product perspective, the progression graph can be imagined as a piece of paper, where the first revisions are situated on the top left and the last revisions are positioned on the bottom right.

In a linear progression graph, the order of revisions indicates that the writer wrote from the beginning of the text straight to the end. Most progression graphs, however, show jagged lines, which indicate jumping back and forth within the text during the writing process (see Fig. 5).

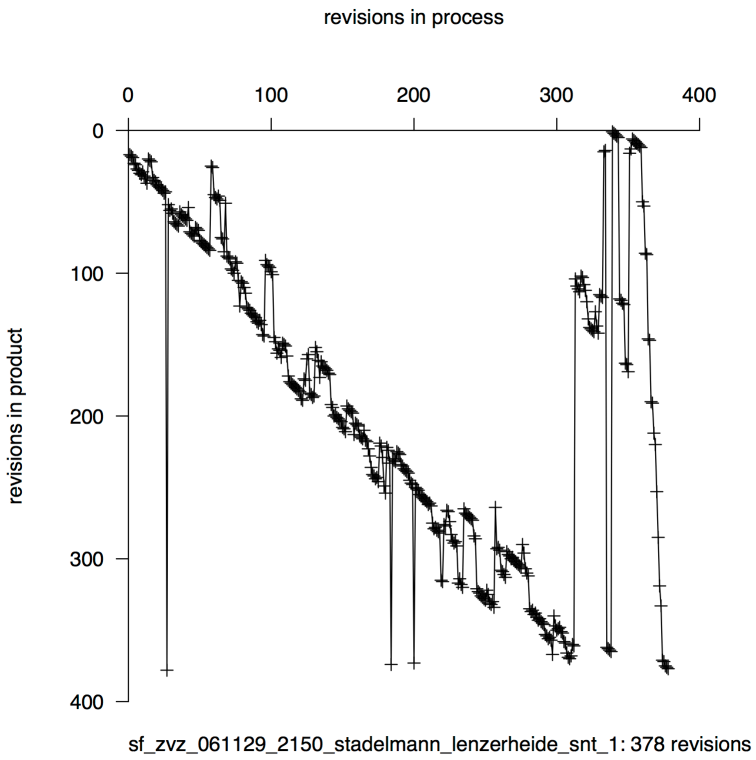

Fig. 5: Exemplification of a progression graph

Drawing progression graphs on revisions as minimal coding units meets the requirements of sociolinguistic writing research that focuses on linguistic practices in context. For neuro- and psycholinguistic analyses of writing processes, progression graphs can also position every single keystroke on axes of time and space. A comparison of such fine-graded graphs with the revision-based variant has shown that the graduation does not generally affect the overall shape of the graph (Perrin, 1997).

2.4.4. Writing phase

A writing phase might be understood as any temporal segment of a writing process. In our understanding, however, writing phases are delimited by changes in dominant revision behavior, for example by a shift from moving forth to moving back or from linear to nonlinear, fragmented writing.

Writing phase: Temporal segment of writing processes that is dominated by a particular writing activity.

Empirically, a writing phase is observable as a more or less homogeneous time series dynamic in the progression data that is delimited by peaks of discontinuity. In terms of Dynamic Systems Theory, the

writing phase is a relatively stable state of the dynamic system between dramatic shifts. The process remains in a specific state, for example the state of formulating a paragraph in a linear top-down movement, until major irritations push it out of this state and it gets attracted by another state in the state space, such as revising a previously written paragraph.

The writing activities that dominate a phase combine material, cognitive, and social aspects:

a) From a material perspective, writing activities are realized through the sequence of revisions over time in the emerging text. For example, formulating often generates a linear writing movement, from top to bottom in the emerging text whereas revising can be identified as writing activity in spatial separated parts of the text. The material aspects of writing activities are reconstructed by computer recordings of the writing process.

b) From a cognitive perspective, writing activities are guided by the writer's intentions, such as goal setting, planning, formulating, and revising. These intentions can be inferred, for example, from the writer's introspections and comments on their writing process.

c) From a social perspective, writing activities can be related to certain contexts and environments. For example, they can take place in an individual setting of one single writer at his or her workplace computer – or in a collaborative setting such as the video editing room, where journalist and editor jointly assemble written language, sound, and pictures.

3. Data and method

In this chapter, I develop the main contribution of this work, namely, method to analyze large corpora of writing process data. First, I show how the data was collected, transcribed, annotated, and merged into a data set ready for statistical analysis. Second, I set out an explanation of how qualitative analysis led to the identification of writing phases. Third I explain how these writing phases were replicated statistically with machine learning methods in order to recognize them in a split-second based on algorithms in writing process data.

3.1. Qualitative perspective: From workplace ethnography to analytical coding

The data of the book beforehand was collected within the framework of the project *Idée Suisse: Language policy, norms, and practice as exemplified by Swiss Radio and Television*. The project was funded between 2005-2007 by the Swiss National Science Foundation. It is part of National Research Program 56, *Language Diversity and Linguistic Competence in Switzerland.*[22] The project investigated how and why the publicly financed Swiss Broadcasting Corporation, in their text production practice, does or does not fulfill the demand of promoting public understanding that is inscribed in the Swiss constitution. According to the constitution, the Swiss Broadcasting Corporation has to promote public understanding primarily across the four languages spoken in Switzerland but also between other social entities, for instance between rich and poor or urban or rural communities (for more details about the aims of the Idée Suisse project, see Perrin, 2013; Perrin et al., 2012).

Pursuing an ethnographic approach, we observed the work of 15 television journalists from three newsrooms by nonparticipant observation. The two German speaking newsrooms, *Tagesschau* and *10 vor 10,* are located in Zurich. *Tagesschau* is the main daily news program with four editions a day, and *10 vor 10* is a news infotainment program with one edition on weekdays. The French speaking newsroom *Téléjournal* was also observed while it produced two paper editions on a daily basis.[23] In all of these settings, we captured the workflow of every journalist over the span of one week, videotaped all editorial meetings, recorded their screens while they were writing, unobtrusively filmed their interactions with video editors, colleagues, and editor-in-chiefs, led initial interviews to grasp their writing biography, and let them comment on one of their writing processes immediately after they had finished writing. Much of this data was transformed, transcribed, and replenished with contextual data, consequently resulting in a multi-perspective corpus.[24]

3.1.1. Building a corpus by ethnographical fieldwork

Gaining access to television editorial rooms for nonparticipant observation requires a considerable amount of preliminary legal clarifications. First, television broadcasting facilities are treated by governments as high security objects because of their importance in crises.[25] Second, classified

---

[22] See (Perrin et al., 2012) for the research design and the results of this project and the Swiss National Science Foundation (2010) for other projects within the framework of the National Research Program 56.

[23] For a detailed ethnographic account of the three newsrooms, their routines and workflows, see Perrin (2013, pp. 9–15).

[24] Evidently, the conditions of journalistic text production have changed since the data collection. This is due in part to the fact that social media gained in importance, technical innovations changed the writing environment, and managerial decisions altered the news production process. However, the scope and aim of this book are based on a representation of writing processes that is less likely to change over time. The methods developed here are applicable to more recent data as well.

[25] Single entry access control systems and electronic badge keys are security measures that are used by both broadcasters.

information, including information subjected to privacy protection or confidentiality of sources may appear at any time on the computer screens we are recording or may be mentioned in editorial meetings and other discursive situations we are filming. To account for the sensitivity of the research setting, data protection agreements had to be signed by every researcher granted access to the data. Third, for security reasons, companies and other organizations are very reluctant to give third persons access to their corporate network which is indispensable for computer assisted recording of writing processes. Furthermore, our on-site researchers had to delete any sequence if the journalist under investigation, the editor-in-chief, or another member of the editorial board wished so. This only happened once. One justification for this high level of trust was the concerted effort in trust building (Perrin, 2013, p. 255) following the principles of transdisciplinary action research that transcends academia (Perrin, 2012) and is research "on, for and with" practitioners (Cameron, Frazer, Rampton, & Richardson, 1992, p. 22).

The 15 journalists were selected by purposive sampling (Patton, 1990), the selection criteria included: similar roles as news editor, different professional socializations and experience, and availability during the period of data collection (see Fig. 6 for an overview of the distribution of age, gender and experience over the 15 journalists of three newsrooms). For these 15 journalists, we recorded all text production processes over a week and collected additional qualitative data by applying the multi-method progression analysis.

		news room														
		Tagesschau					*10 vor 10*					*Téléjournal*				
Journalist	Name	ST	ES	CP	KR	HS	SE	MP	MR	CB	MK	JR	RG	OK	CS	CA
	Born in 19..	48	46	68	71	47	64	67	73	70	79	72	59	71	76	52
	Gender	M	M	F	F	M	F	M	M	M	M	M	M			
	– wires				5	9										
	– print	3	16	2	1	6		5	3			3		7		8
	– radio	5					2						20		5	
	– online											1				
	– television	26	15	9	1	14	13	10	7	1	6	3	3	9	2	8
	– total	34	31	11	7	29	15	15	10	1	6	7	22	16	7	16

Fig. 6: Journalists' professional experience in years

Multi-method approaches consider the object of study from several perspectives and thus provide more dimensions to reconstruct text production than single-method approaches (Beaufort, 1999); (Sleurs, Jacobs, & Van Waes, 2003); (Dor, 2003); (Perrin, 2006). Progression analysis (Perrin, 2003), an ethnographic, computer-based, multi-method approach obtains data on three levels: a) work situation, b) writing movements, and c) writing strategies:

a) Before writing begins, details about the work situation are elicited via interviews and participatory observation. During writing, movements are measured with computer-based recordings. After writing, the repertoire of writing strategies is deduced with data-supported retrospective verbal protocols.

b) During the writing process, progression analysis records every writing movement. In the larger investigations with progression analysis, the logging programs run behind the text editors that the writers usually use, for instance behind the user interface of the news editing systems. The logging follows the writing process over several workstations and does not influence the performance of the editing system. It records all keystrokes and mouse movements as timed actions related to text entities and writer identification.

c) Writing strategies focus on the writer's reinforced, conscious, and therefore articulable ideas of how decisions are to be made during the act of writing, so that the writing process or text product has a great probability of taking on the intended form and fulfilling the intended function. This level of progression analysis was used to validate the quantitative methods (see section 3.3).

If the workflow of the journalist under investigation allowed, we started the week of non-participant observation with a one hour guided interview to grasp the writer's biography and her or his socialization as a writer and journalist. These interviews were coded by propositional analysis and revealed parts of the journalist's, the organization's, and the news room's medialinguistic mindset (Perrin, 2013, p. 69). By propositional analysis, writing strategies are deduced from the verbalizations.

Writing strategies that are coded in the propositional format have the form *to do x as a function of y*, se.g. *to do x, because y is true*, or *to do x to achieve y* (Perrin, 2013, p. 55). An example of a writing strategy, based on the first utterance in the protocol in Example a, would be *take out words to shorten the text*. In the nomenclature, we call this data type (see Fig. 7).

Where?		When?		Who?	What?	How?		
Broadcaster	Newsroom	Date YYMMDD	Time hhmm	Author	Topic	Data type	Session x	Format .xxx
télévision	téléjournal	070219	1245	kohler	roadpricinglondon	desktop	1	.avi
Suisse	(5 journalists)	…	…	…		verbal protocol		.avi
Romande						review protocol		.avi
Schweizer	tagesschau	061106	1300	scheben	nicaragua	s-notation	1–2	.htm
Fernsehen	(5 journalists)	…	…	…		mid-syntax		.txt
						phases		.csv
						progression		.pdf
						score		.pdf
	zehn vor zehn (5 journalists)	061128	2150	stadelmann	kabelknatsch	text	1	.txt
		…	…	…		item		.mov
						item-context		.txt
		070219	0930	editorial		discourse		.mov
		070219	1045	kohler		frame		.mov

Fig. 7: Data types and filenames[26]

[26] Where the name is long, the underlined characters form the filename whose delimitation with a underscore allows for automated processing. Thus, the three files that serve as examples in this table are: tsr_tj_070219_1245_kohler_roadpricinglondon_desktop_1.avi, sf_ts_061106_1300_scheben_nicaragua_snt_1.htm, sf_zvz_061128_2150_stadelmann_kabelknatsch_text_1.txt

Due to the journalists' workflow, it was not always possible to initiate the nonparticipant observation with the guided interview because they had either to conduct desk research, undertake site visits to get footage for their news items, or immediately start writing. In the latter case, we started the data collection by recording their writing processes with a screen recording software.[27] We accessed the screens of the journalists over the local network using screen mirroring. This procedure had three advantages: First, we did not have to install the screen recording software on all computers that were potentially used for text production and consequently only needed one license for the mirroring software (Techsmith, n.d.). Second, the journalists were accustomed to this unobtrusive procedure because they used screen mirroring as well when they encountered a computer problem and needed help from their IT staff. Third, the data management was more feasible because all screen recording data were saved and managed on one hard drive. This data type is referred to in the nomenclature as *desktop* (see Fig. 7).

Again, by purposive sampling we selected one writing process per journalist to record a cue-based retrospective verbal protocol in order to grasp the writer's individual language awareness. Adapting Svalberg (2007), Perrin (2013) defines language awareness as the "consciousness and attentiveness in solving language problems in specific communication situations" (p. 38). Practically, we let the journalists comment on the chosen writing process immediately after they finished writing. *Cuebased* implies that the journalist commented freely on what happens on the screen. In some cases the researchers initiated the comments of the journalists by asking questions such as: What are you doing here? Why did you do that? Proceeding in this way, we do not postulate that we capture the actual reason for the writing behavior, but only a conscious reason uttered spontaneously that could have affected their writing in this situation and could also affect it in other, comparable situations.[28] In the nomenclature this data type is named *verbal protocol* (see Fig. 7).

The data types *s-notation*, *mid-syntax*, *phases*, *progression* and *score* are writing process data. *S-Notation* files include the revisions as described in section 2.4.2. They were generated from *midsyntax* and reconstruct the writing process in the ordinal format of movements, insertions, and deletions. To add contextual and qualitative information, the mid-syntax was transformed into the comma-separated value format that statistical software can access (see section 3.2.1). This filetype is named *phases*. The visual manifestation of writing processes, i.e., the above described progression graphs (see section 2.4.3), is named *progression*. The triangulation of the writer's strategies, qualitative insights into the text production, and the ethnographic reconstruction of the work situation is visualized in a progression *score* (see section 3.3). For an overview of this five writing process data types see Fig. 7 .

Whereas the cue-based verbal protocol is very specific to one writing process (because we did not ask summative questions), the subsequently led standardized *review* protocol (see Fig. 7) records answers to identical questions for all 15 writing processes with verbal protocols. Questions about aim, scope,

---

[27] We used the paid software *Camtasia* (Techsmith, n.d.). *OBS Studio* (Open Broadcaster Software, n.d.) is a very versatile and free open source alternative.

[28] The veridicality of retrospective verbal protocols is limited by the constraints of memory: Writers might forget the reasons for their decisions or make them up (Levy, Marek, & Lea, 1996). The alternative, called concurrent protocols or think-aloud protocols, on the other hand, are subjected to reactivity (Janssen, Van Waes, & Van den Bergh, 1996; Stratman & HampLyons, 1994). Experiences in the field show unsurprisingly that journalist do not always remember the reasons for their decisions, but surprisingly often – for them – they do. As an analogy, a sequential process can be considered to be like a chess game where the players manage to remember a whole game by replaying or rethinking it move by move, but could not recall one move without the sequential order of the previous or the following one. However, concurrent protocols are not an option in professional settings, especially not in editorial rooms where "speaking during writing would make the writers slow down and irritate the colleagues around them in the open-plan offices" (Perrin, 2003, p. 916).

and preliminary products of the news item have been asked (for the questionnaire see section see section 7.3).

The writers' shared language awareness has been captured by propositional analyses of the journalists' discussions with colleagues in the editorial room, at editorial meetings, with video editors during collaborative news production processes, and with speakers and anchors before and after broadcasting the news item. We named this data type in the nomenclature of filenames as *discourse* (see Fig. 7).

The final products of the writing processes also belong to the corpus. The news item as a video file is named an *item*. The transcript of the video file is named the *text*. The collected news items were designed for a Swiss-German audience, and as such we added contextual information for other audiences in the data type *item-context* (see Fig. 7).

Combined, this data allow for an in-depth and dense reconstruction of the writing process, including the semantic level of the news items, i.e. the content. While writing, journalists make numerous decisions based on the topicality of their news item. This topicality is not unique, since other news items have similar topical structures and comparable narrative or genre structures are used to display them (Russell, 1997; Wrobel, 2000). This renders the comparison of writing processes challenging: A researcher needs general and abstract categories for comparison, but the journalist has to consider the specificity of the item's topic to produce a differentiated news item.[29]

> By abstracting from 'real' time (absolute time scale) and from the particular text content (words and phrases), we can eliminate 'nuisance' effects[30] that might otherwise mask the most interesting writing phases. In analyzing large corpora of field data, we are not interested in the content of a particular text (the meaning of the words and phrases used) or in the particular text-producer (the author) but rather in the progression patterns as traces of writing phases. (Perrin et al., 2011, p. 3).

However, in section 3.3 I elaborate on how to combine the qualitative and the quantitative perspective, but first I show how the qualitative data has been analytically coded to make it accessible for quantitative analyses.

3.1.2. Transforming qualitative into quantitative data by analytical coding

Building on the phase concept defined in section 2.1.4, phases had to be identified that are observable as more or less homogeneous revision patterns in the progression data and that are delimited by peaks of discontinuity. Benefiting from the interdisciplinary architecture of the *Modeling writing phases* project, the two teams, linguists and statisticians, started independently by applying their discipline's methods and procedures to discern writing phases. In a second step, the results were discussed and methods and procedures were refined. This process was repeated numerous times until the writing phases presented in section 4 were identified and those of section 4.2 modeled.

The statisticians treated the writing process data as times series data. The inherent temporal characteristics of writing process data render them well for times series analysis: "A time series is a time-oriented or chronological sequence of observations on a variable of interest" (Montgomery,

---

[29] On the other hand, acknowledging the uniqueness of a topic conflicts with another task of the journalist, namely, reducing complexity and consequently showing not only what is specific for this news item, but also what is similar to others.

[30] "Typically, text production is subject to perturbations (phone calls, travel, idle periods, etc.) which contaminate absolute time" (Perrin et al., 2011, p. 3). A number of these perturbations might have an effect on the writing process, although with various effects that can be accounted of only in focused single case studies.

Jennings, & Kulahci, 2015, p. 2). The time series data format allows for other statistical analysis than cross-sectional data, which is generally the data format generated by polls and questionnaires. The statisticians started with a measure which is associated with the linearity of a writing process and which can identify a so-called “global break-point” in the underlying time series dynamics. Consider the following progression graph (Fig. 8):

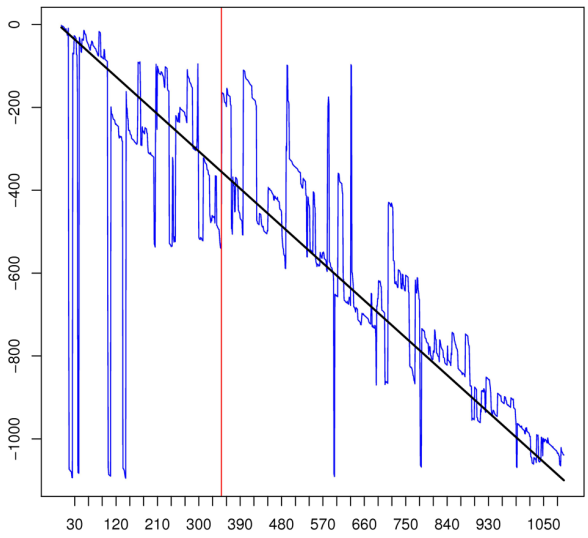

Fig. 8: Deviation from linear trend

In Fig. 8, the jagged blue line shows writing movements, i.e. the revision patterns of a writing process. Revision 29, for example, is situated at the bottom of the final text product, but took place at the beginning of the writing process. In other words, the 29th of over 1050 revisions ended up being the last sentence of the finished product – a fact that strikes as somewhat counter intuitive to an assumed linear writing process. The diagonal black line shows the linear trend, the average progression of the blue line. The vertical red line marks the extremum of cumulated deviation (to be explained after Fig. 10). After “de-trending”, the progression graph appears as follows (Fig. 9):

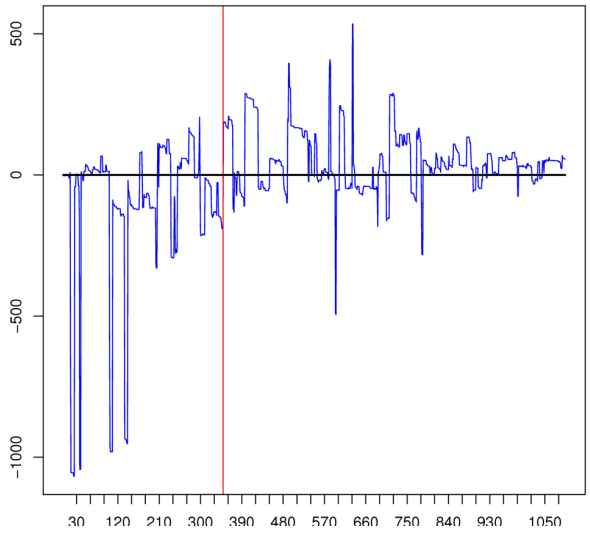

Fig. 9: De-trended progression graph

The horizontal line in Fig. 9 corresponds to a perfectly linear writing process of an individual who writes a document consecutively from the first to the last character, meaning without jumping back and forth in the text produced so far. Obviously, such linearity is not true of the writer observed here. The deviations from the horizontal line are an interesting case to analyze. The cumulated sums of these deviations are shown in Fig. 10, below.

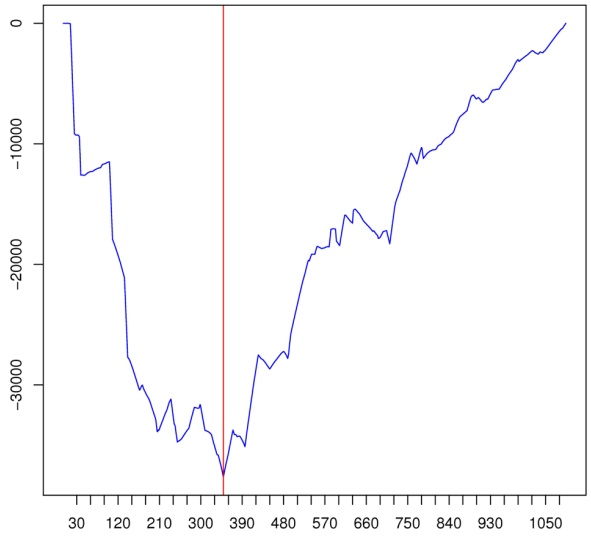

Fig. 10: Cumulated deviations from linear trend

In Fig. 10, the jagged blue line indicates cumulated deviations of the progression graph from its linear trend. For each time point on the abscissa, the y-value on the ordinate corresponds to the sum of all deviations in Fig. 9 up to this time. As can be seen, this sum has an extremum indicated by the central vertical line (which has been reported in the preceding two figures Fig. 8 and Fig. 9). From the beginning of the writing process, the writer increasingly deviates from the purely linear writing process until the extremum of the cumulated sums is attained. Once that time point is exceeded, the writer progressively returns to the horizontal line, i.e. the linear process. The position of the extremum, its value as well as the shape (asymmetry) of the above time series on both sides of the central vertical line, can be used to describe the writing process and compare it with others.

Linguists started to discern writing phases by inspecting the progression graphs for homogenous revision patterns and peaks of discontinuity, e.g. changes from linear to less linear writing and back. Then they verified their findings with the qualitative data. Following the principles of abduction (see section 2.3), the classification of writing phases has been developed in an iterative, i.e. path-dependent process. In practice, two researchers classified writing phases in several sessions alternating between together and alone, and discussed their results and to refine their classification. This time consuming

and patience demanding procedure bears the advantage of getting to know your data and your reflections on them thoroughly.[31]

Consequently, the writing phases presented in section 3.1 are the result of hundreds of iterative decisions and evidently, it could happen that other researchers applying the same procedure could yield other phases, which is not desirable in terms of reproducibility. But in contrast to the numerous concepts of writing phases discussed in section 1, they are empirically grounded on over one hundred writing processes. Furthermore, it is not that far-fetched to believe that other researchers could come up with a similar concept of writing phases – after having abductively tried hundreds of others.

3.2. Quantitative perspective: from descriptive statistics to machine learning methods

In the previous section I described how the data was collected in an ethnographically informed way and then transformed into a format that allows for statistical analysis. In this section, I introduce the data format of the investigated writing processes and the statistical procedures that were used for the analysis.

3.2.1. Rendering writing process data accessible to statistical analyses

Applying statistical methods in writing process research has a long tradition, especially in the strand of cognitive psychology (see section 2.2). Statistics have been used to quantize such different aspects of writing processes as fluency (Abdel Latif, 2013; Chenoweth & Hayes, 2001; Flower & Hayes, 1981), the relationship between written encoding and psychological, demographic and stylistic variables (Ruffner, 1981), and – with the highest number of publications – the effect of different teaching methods on students' writing, mostly in the form of randomized intervention studies (for an overview see MacArthur, Graham, & Fitzgerald, 2016). Furthermore, most of these studies investigate the writing process via the writing product, i.e., the final text, and questionnaires or interviews.

But if the approaches are narrowed down to those that use empirical datasets of writing processes in natural and non-experimental settings, and that include all revisions of the analyzed processes and are not simplified, i.e. categorized versions, not many studies remain. The lack of empirically grounded research in natural settings has been often criticized. For example, in. the critical part of Janssen's (2007) review of the edited volume *Written documents in the workplace* by Alamargot, Terrier, and Cellier (2007):

> Most chapters are of a theoretical, philosophical nature and – in my view – do not yet provide the empirical evidence necessary to support the claims. Moreover, only a few articles factually present original data. While I realize that there can be different ways of pursuing knowledge, I myself am a strong believer in empirical data. The models presented in this book really deserve to be tested in experiments, multiple case studies, corpus studies, etc. In addition to thinking and theorizing about writing and reading in organizations, it can be challenging to

[31] Especially in social sciences, but also in other disciplines, a test for inter- or intracoder reliability is often used to prove that another coder (intercoder) or another coding session of the same coder (intracoder) leads to the same result of coding or classification as the first coder or session. One effect of this validating procedure can be that few reproductive studies are made because the authors already proved via the tests that a replication would come to same result. Another effect can be that the so constructed and validated, classification systems gain in reproducibility but do not grasp a lot of complexity. However, the applied ethnographic approaches consider the iterative and cyclic coding as a helpful device to get to the data thoroughly (see section 2.3).

actually study readers and writers in context. The contributions in *Written documents in the workplace* present many interesting hypotheses that may very well be tested in organizational or more controlled situations. (Janssen, 2007, p. 86)

One reason for the lack of empirical studies on writing processes in non-experimental settings at the time Janssen wrote his review, was the technical challenges.[32] At the time, the available software for keystroke logging was rather buggy and time consuming to handle and for reasons of obtrusiveness and reactivity not suited to record writing processes in non-experimental settings. Keystroke logging software for experimental settings need the writing to happen within their environment, as is the case for *Scriptlog* (Andersson et al., 2006; Strömqvist, Holmqvist, Johansson, Karlsson, & Wengelin, 2006) or *Translog* (Jakobsen, 2006). However, journalistic text production often takes place in server-based news editing systems, and television journalists do not only type their texts within these systems, but also search the news agencies and video databases, conduct basic video editing, create the captions, link the video sequences to the intended position in their texts, and so on. Consequently, a substantial amount of the logged keystrokes typed during a text production process are never intended to be part of the final text. *Inputlog* (M. Leijten & Van Waes, 2013) is able to log keystrokes independently of the program they are typed, but for a long time it remained a tedious task to differentiate the inputs in the various programs (word processor, browser, mail client, editing system, messengers, and many others) in order to reconstruct the revisions of a selected writing process. Furthermore, the software has to be installed and managed on every workstation. To overcome these technical barriers, a scholar of writing processes in complex natural settings has to either code or let code a customized software or to manually reconstruct the writing processes based on screen recordings (Perrin, 2013, p. 256).

We recorded the writing processes in the MID-syntax. The three letters stand for *move*, *insertion* and *deletion* (see Ex. 2 for an exemplification).

[32] Another, non-technical reason for the small number of empirical studies on writing processes in professional settings is the personal and ideological gap between academic research and the professional world. Workplace ethnography and writing research in professional settings requires a substantial amount of time and money from the partner in the field. This commitment is hard to achieve without previous trust building and personal relationships, in particular with the decision-makers in the organization's management (Perrin, 2012b).

```
[…]
2.0 I 11 2.7 Ils sont do
0.2 I 2 0.1 nc
62.4 I 26 9.2  venus de tout le pays par
13.8 I 4 1.9  dez
11.4 D -2
1.4 I 8 2.8 izaines
5.6 I 11 2.7 de milliers
6.7 I 5 1.8 ....
14.3 I 50 16.9 Par la route et même pour certains par la voie exp
17.1 I 17 7.4 ress de la Médite
14.4 D -1
0.8 I 7 4.0 érannée
[…]
0.0 I 16 5.8 Ils sont venus p
28.6 M 53
1.0 D -7§
2.5 I 10 2.0 tranquille
20.0 M 25
1.5 I 38 14.3  Point commun de tous ces manifestants
16.3 I 12 2.0 , le drapeau
[…]
```

Ex. 2: Exemplification of the mid-syntax on tsr_tj_070214_1245_guillet_libanon_mid_3

The first number from the left on every line shows the time in seconds that has passed since the last revision. This timespan was not interpreted for the reasons described in section 2.4.2. The second element of the MID-syntax depicts the kind or revision the writer has made: an insertion, a deletion, or a move to another position in the text. For S-notation, successive revisions of the same type are combined into one because they only occur if a pause interval is defined. The third number delineates the number of characters that have been inserted, deleted, or moved. The sign of this number reveals if the move or deletion was made forwards or backwards from the last position of the cursor. The fourth element from the left, if the revision is an insertion, shows the duration in seconds, which was also left out of consideration in the concurrent research. Finally, the logged keystrokes follow.

To enrich the basic MID-syntax with additional data points, such as the qualitative coding of writing phases, the 120 writing processes were converted into the comma-separated value format, which is accessible to statistical software and can be depicted as a table (see Ex. 3).

revision_nr	position_nr	hor_lines	ver_lines	jumped_rev	phases	start_stop
1	9				dancing	start
2	10	44		0	dancing	

Ex. 3: Exemplification of the comma-separated value format
Source: sf_zvz_061123_2150_stadelmann_bauernsterben_snt

The first two columns contain the number of the revision in process (*revision_nr*) and the number of the revision in product (*position_nr*), which are also used to draw the progression graph (see section 2.4.3). The next two columns are used to mark specified sections of the product (*hor_lines*) or the process (*ver_lines*) in the progression graph with a horizontal or vertical line. The fifth column indicates, for every revision, how many revisions in the product have been jumped over (*jumped_rev*) since the last revision. This number is needed to model writing phases because it is an indicator for jumps in a writing process (see section 3.2.2). The penultimate column tells which writing phase (*phases*) was coded for the revision on this row, and the last column is an auxiliary dichotomous variable, used to tell at which revision the specified phase starts and stops.

Ex. 3 represents the basic writing process data set that was expanded with additional qualitative and quantitative data. Whereas some qualitative data has to be coded first and then entered manually, additional quantitative data has been generated by applying statistical procedures to this basic dataset. In the next subsections, I elaborate on the statistical procedures used to model writing phases.

3.2.2. Modeling writing phases

Once the linguists and the statisticians had agreed on the writing phases to be presented in section 4.2, the next task for the statisticians was to model these writing phases so that they could be automatically identified within large writing process corpora. Modeling serves to reduce the complexity of real data to render it easier to understand and work with. Depending on the amount of data and the kind of processing, modeling can be approached from either an epistemic or a statistical angle.

Epistemic models take theoretical ideas based on specific empirical knowledge and streamline them into simplified structures, for example, Perrin's (2013) model of situated newswriting or Flower's and Hayes' (1981) cognitive process model.[33] An epistemic model may take virtually all qualitative information into account but the amount of information is limited by the processing capacity of the human brain.

Statistical modeling, on the other hand, reconstructs the structure of given data using algorithms. Much like epistemic models, models in the statistical sense are a simplification of reality but they can process significantly more data and identify structures that would otherwise remain obscure. The use of a statistical model limits the investigation to a chosen dataset but may surpass the data processing capacity of the human brain by a factor of millions. Hence, one of the biggest challenges of statistical modeling is to separate what is relevant from what is not – to separate signals from noise.

The first step in mastering this challenge is choosing and defining a model that best fits the data. In the *Modeling writing phases* project, the statisticians Beate Sick and Marc Wildi decided on a random forests classification model, a machine learning method that automates analytical model building (Wildi, 2007). The term *random forests* was coined by Breiman (2001), building upon Ho's (1995) insights.[34]

In essence, random forests models are decision tree models. These examine how classifiers apply to the data in a hierarchical process. Applied to writing process data, they identify the writing phase to which a revision or a group of revisions belongs. To this end, features, or "predictors", that describe specified characteristics of the data are extracted and incorporated into algorithms. They contain aggregated information on the variables presented in section 3.1.2 above: information as to how successive a specified revision behaviour is, how many revisions are jumped over, in which direction, and so on. The features differ in scope: some of them classify only one revision, others apply to groups of different sizes (see the R-scripts for feature creation in section 7.4.1 in the appendix).

Both decision tree models and random forests models apply features to a bootstrap samples of data. Thus, each tree is constructed with varying but overlapping subsets of data. In practical terms, each tree in Fig. 11 below is trained with 63.2 percent of the data, and cases are drawn at random with replacements from the original data. The remaining 36.8 percent of the data is used to calculate the

[33] See section 2.2 for a description of both models.

[34] Breiman (2001) used the plural "forests." The plural corresponds to the idea of the model that numerous forests are calculated, not one.

misclassification error, called the *out of bag (OOB) error rate*. The OOB rate indicates how well each random forest performs on data it was not trained with.

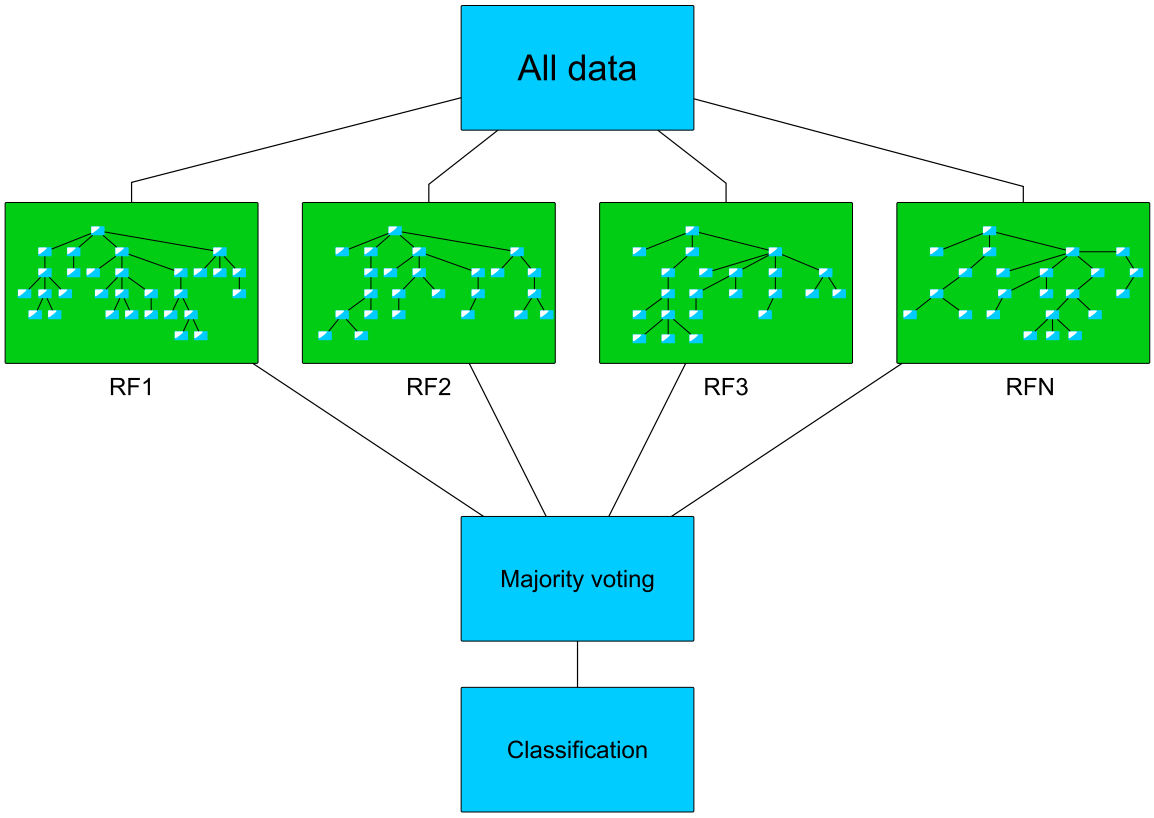

Fig. 11: Random forests model

The difference between standard decision tree models and a random forests models lies in the choice of features used to classify the data at each node of the tree:

> In standard trees, each node is split using the best split among all variables. In a random forest, each node is split using the best among a subset of predictors randomly chosen at that node. This somewhat counterintuitive strategy turns out to perform very well compared to many other classifiers, including discriminant analysis, support vector machines and neural networks, and is robust against overfitting. (Liaw & Wiener, 2002, p. 18)

With this in mind, not only is the data used to grow random forests selected at random, but so too are the classifiers.[35] Finally, each forest votes for a classification, and the classification is chosen by majority vote.

All of the revisions in all of the writing processes have been classified. One can imagine this process as a classification window gliding through all writing processes and classifying each revision based on the

[35] The classifiers and their performance are topicalized in the result section 4.5.

random forests. In the next section, I will discuss how this quantitative approach was combined and validated with the qualitative data.

3.3. Combining methods

After the linguists and the statisticians had analyzed the data separately, it was the challenge – and the pleasure – to bring the two different insights together. Whereas the macro indices, for example the cumulated deviations from linear trend (see section 3.1.2), were found to be helpful for rapidly detecting critical situations in writing process data, the modeling of writing phases had to be validated on two fronts. First, a confusion matrix was needed to verify if the phases the random forests model identified corresponded to the phases the linguists tagged manually (see section 4.1). Second, how the identified phases and their meaning relate to the other qualitative data was cross-checked. For this, the progression score and other qualitative data were used.

In this chapter, I introduced the terminology and the methods used throughout the rest of this book. First, I depicted how the writing process data was ethnographically collected at news desks of Swiss television journalists and enriched with contextual data. Second, I showed how the data was qualitatively analyzed to identify writing phases. Third, I depicted how I applied structure detection methods to model writing phases and how I analyzed the effects of their sequence. Finally, I discussed the combination of the qualitative and quantitative methods and their findings.

4. Results

As described in section 3.1.2, the linguists discern writing phases by inspecting the progression graphs (see section 2.4.3) for homogenous revision patterns and peaks of discontinuity, i.e. changes from linear to less linear writing and back. Hence, the progression graphs are described by looking at the position of the dots on the graph relative to each other.

The relative positions of two successive dots on the progression graph that are separated by previously made revisions, which we call a "jump," are particularity relevant. Valuable numbers for discerning writing phases include the size of such jumps (big jumps appear as spikes on the graph, small jumps as a straight line), their direction (forwards or backwards), the number of changes in direction, the number and size of spikes in the graph, and the distribution of the size of jumps on the graph.

A writing phase – defined as temporal segment of writing processes that is dominated by a particular writing activity – may be defined on various levels. A particular writing activity can be observed for a small group of revisions, for example correcting a typo or adjusting grammatical forms as consequence on adjustments on the syntax level, or for larger groups of revisions, for example adjusting the whole news item because a correspondent delivered a new quote that calls for a different dramaturgy. Hence, the size and function of writing phases differ according to what is understood as particular writing activity.

In order to account for different levels of writing phases, Severinson-Eklundh and Kollberg (2003) distinguish the following three levels of what they term "revision episodes":

> Type 1: Repetitive revisions at one cursor location
> Two or more immediate revisions are made at one cursor location, where the writer is currently producing text. This episode type occurs, for instance, when the writer is trying out different words in one place in the text to find the right way to express something, deleting and inserting repeatedly at the same position.
>
> Type 2: Embedded revisions
> This episode type includes all cases when one revision is carried out during the course of another revision. It occurs when the writer is making an insertion which is then modified before it is finished, for example, by substituting a word in the sentence being inserted within the text.
>
> Type 3: Sequence of revisions in previously written text
> The writer interrupts the text production to make a sequence of revisions at different locations in the text written previously. The revisions may or may not be semantically related to one another. After the last revision in the sequence, the writer either resumes writing at the position of the interruption or ends the writing session. This type of episode occurs, for example, when the writer goes through a paragraph just written and makes a number of revisions in it. (pp. 872 – 873)

The above definitions and typology of *revision episodes* leave many kinds of segments in writing processes unclassified and therefore uninvestigated. Writing down a paragraph without revisions, for example, is not covered. As these segments beyond revisions (and episodes or chunks) are relevant phases too, they have to be considered on higher levels of the analysis.

As a consequence, I differentiate four levels (and numerous types) of writing phases:

Phases on the chunk level (see section 4.1)

Phases on the turn level (4.2)

Phases on the run level (4.3)

Phases on the session level (4.4)

4.1. Phases on the chunk level

On the chunk level, phases consist of sequences of linear revisions, i.e. revisions that take place one after the other in both time and space. These phases are delimited by discontinuities in the small-scale revision activity, typically around words or phrases. A phase shift on this level can take the dynamic system of writing from moving forwards in linear text production to stepping back in order to alter single formulations. This shift is where a new chunk begins.

Writing chunk: Segment of the writing process in which two or more revisions are performed in a strictly linear sequence.

Within a chunk, writers can execute revisions immediately when writing, without moving the cursor back and forth. Such chunk-internal revisions occur for example when writers correct typos, change grammar markers or alter lexical choices on the go or while continuously writing their text down. Chunks are terminated when writers move the cursor back in order to change previously written text.

On the next higher level of the multilayered phase model – the turn level – chunks represent the absolutely linear strictly walking movement – a rather exceptional variant in a typology of movements. The principle of linearity itself scales up to continuity, i.e. to various types of continuous movement through parts of the emerging text

4.2. Phases on the turn level

On the turn level, phases consist of continuously producing or revising wider text parts, typically sections of texts. These phases are delimited by discontinuities in the mid-scale revision movement and activity. In the newsroom, a phase shift on this level can take the dynamic system from writing paragraphs in a close-to-linear movement to the more fragemented movement of skipping from subheading to subheading in order to revise them.

Writing turn: Segment of writing processes that is characterized by a specific type of revision movement, delimited by changes of movement.

I identify five types of turns. They are dominated by revision movements through emerging text parts: walking (see section 4.2.1), dancing (4.2.2), skipping (0), jumping (4.2.4) and unclear (4.2.5).

On the next higher level of the multilayered phase model, walking, dancing, and skipping down through the entire emerging text result in a run. The principle of continuity itself scales up to consistency, i.e. to various combinations of turns adding up to a run (see section 4.3).

4.2.1. Walking turn

Walking turns are dominated by small step-by-step movements. When walking turns are observed on the screen, the text evolves on screen as if a computer reproduced a previously stored coherent text part character by character – interrupted only by occasional corrections of typographical errors.

Walking: Writing movement in which the writer proceeds from top to bottom, in a linear sequence of revisions.

In the progression graph of a perfect walking turn, there is a straight line from the upper left to the lower right (Fig. 12). Each revision in the walking segments of a progression graph (except for the last revision) is followed by a revision that is the next one in the final text.

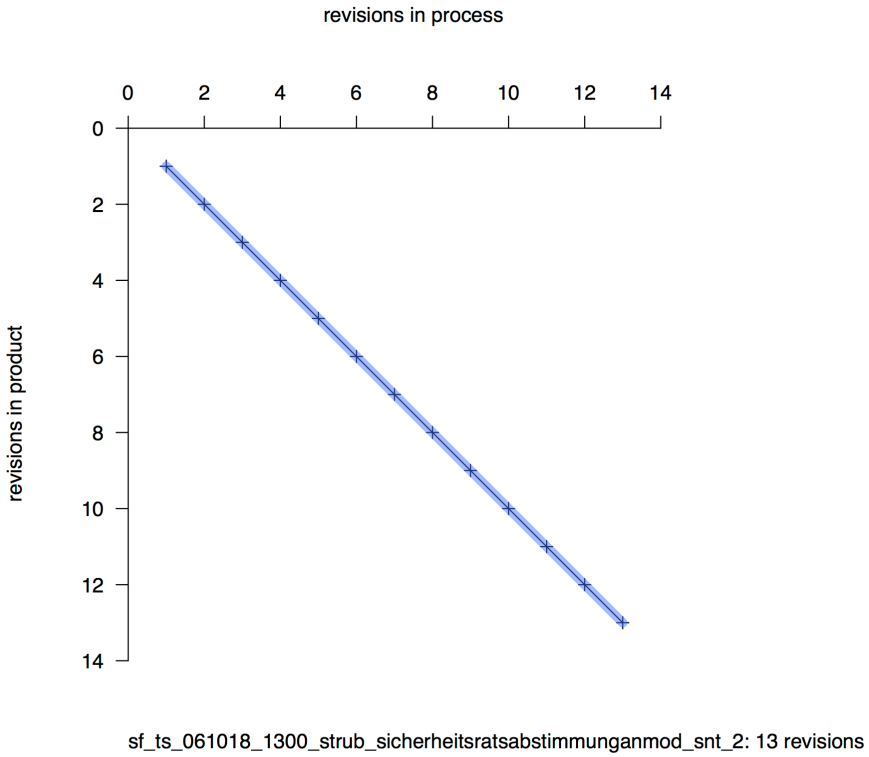

Fig. 12: Phases on the turn level: walking

In most analyses, however, segments of progression graphs that show a few minor jumps are also classified as a coherent walking segment if the jumps include only a few revisions and if they interrupt long linear stretches of the progression graph. In the graph in Fig. 13 look as if they interrupt the long initial walking turn highlighted in blue. However, this is only due to the construction of the progression graph: If the two last revisions would not be part of this writing process, the long initial walking turn would not be graphically interrupted. Hence, later revisions have an effect on the visual presentation of previous revision patterns.

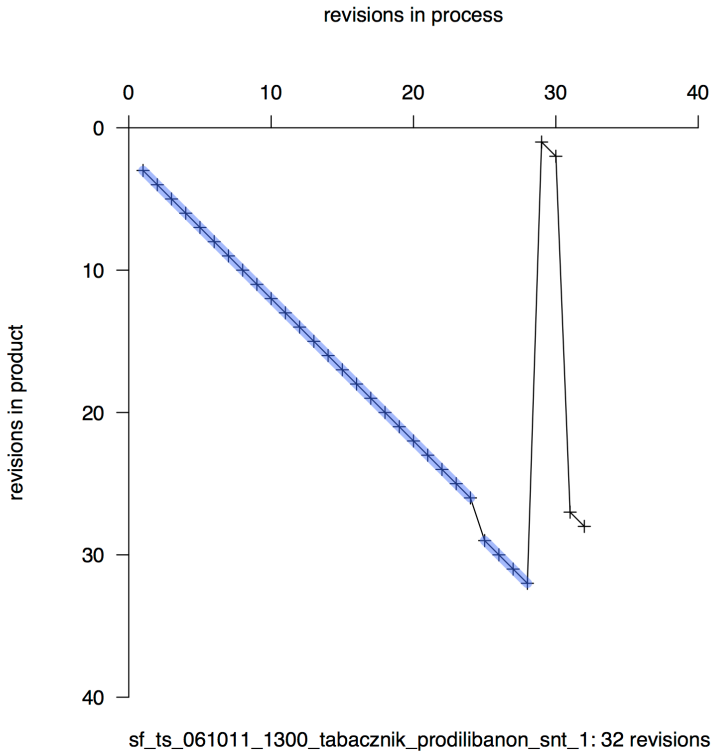

Fig. 13: Seemingly interrupted walking turn

From a mental perspective, the material activity of walking can be interpreted as formulating: as writing down coherent thoughts in the reading order and immediately revising typos. The results of the qualitative analysis (especially of the data type *desktop*, i.e. the screen recordings) show that walking turns have various causes and functions. First, they are an indicator of note taking. Taking notes in a linear way, making only minor typographical corrections, for example while watching or listening to a live feed of a media conference results in walking turns. Second, walking turns can be manifestations of graphomotoric habits, capabilities, and disabilities. If a writer make a lot of typos, but immediately corrects them, then long walking turns will be produced.

For example, E.S., an experienced writer with self-reported dyslexia, immediatly corrects his numerous typos, which results in an above average number of walking turns in his writing processes (see Fig. 29 on page 58). The same is true for S.T., an experienced writer whose typing speed is considerably slow because he mostly uses only his two index fingers to type. S.T. provides an astonishing case for mental pretext, i.e. mentally stored, and in his case rather concrete, text versions. His unusually slow typing style for a professional writer might carry the consequence that he is able to develop a very concise and detailed text in his mind while writing out a story. This scenario was observed on a site visit observing the production of an item about a fair farming[36] where he spoke with numerous people and visited multiple locations without taking a single note. He and his camera man traveled to the location with a broadcasting van so as to be able to produce an item for the noon edition of *Tagesschau* without returning to the newsroom. Twelve minutes before his item was about to be sent, he started writing – such a short time span was not observed for any other journalist responsible for producing an entire news item. For S.T., however, this behavior is the rule rather than the exception. Due to his detailed mental pretext, he achieved his goal effortlessly and without any observable signs of haste.

[36] sf_ts_061012_1300_tabacznik_olmaschmid_item

These walking revision patterns also emerge if the writer has fallen into the habit of consistently mistyping certain words, but then incorporated the immediate correction of the misspelling in the graphomotoric memory as well. This cause for walking turns can be related to the applied typing system, be it a touch system, a two-finger system, or a hybrid system.

Nevertheless, walking turns are principally indicators for a writing behavior that produces text in a forward oriented way. If more backward oriented revising takes place, a walking turn becomes a *dancing* turn (see the next section). This is reflected in the feature creation that describes a walking turn by an algorithm in order to recognize the walking turn in writing process data (see R-code in section 7.4.2, lines 398 – 505).

4.2.2. Dancing

Dancing turns are dominated by oscillating movements. When dancing turns are observed on the screen, the text evolves as writing alternating with local revising of the text just written.

Dancing: Writing movement in which the writer proceeds from top to bottom but keeps going back to revise formulations just written.

In the progression graph of an ideal dancing turn, there is an oscillating line from the upper left to the lower right. In most analyses, however, dancing segments include short walking-like sequences and single mid-range jumps (see the revisions highlighted in green in Fig. 14).

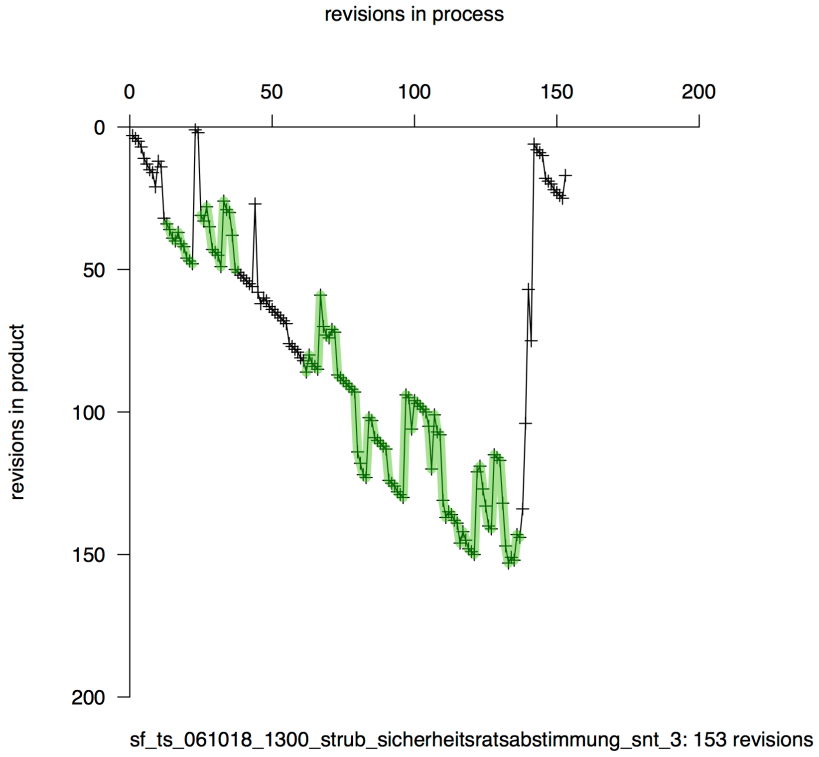

Fig. 14: Phases on the turn level: dancing

From a cognitive perspective, the material activity of dancing can be interpreted as formulating – as writing down coherent thoughts in the reading order and immediately revising the formulations of those thoughts. While dancing is principally a quantitatively productive revision behavior, the qualitative analysis also shows that, in some situations, dancing turns can be a sign of extended revising on a

comparatively small text part, which thus slowed down the overall text production. Ex. 4 shows how the journalist S.E. wrote the first paragraph of an item about the reasons why the number of farms in Switzerland are decreasing. As presented in section 2.4.2, the underlined characters represent the final text in S-notation, thus S.E. reformulates the two sentences of this paragraph several times.

```
1{Meinrad Etterlin 2[arbeitet als |2]2,12{war Bauer 29[und ]29|30 25[bes13[s|13]13ass
einen ]25|26 26{mit }26|27eigene27[n]27|28 28{m}28|29 Hof. Vor fünf Jahren gab er14[ih|14]14
ihn auf, seitdem 21[a15[re|15]15,16[b|16]16rbeitet er ]21|22}12|17 22{hat er einen
Job }22|23 17[ist 5{seit 5 Jahren }5|6Gemei3[dn|3]3ndearbeiter in 4[Me|4]4der Aargauer
Gemeinde Merenschwand.6[Vor 5 Jahren war er noch Bauer. |5]6 Vorher ]17|18 18{als
Gemeindearbeiter 23[in der Aargauer Gemeinde ]23|24 24{im
Dorf }24Merenschwand19[. ]19|20}18|19 20[war er Bauer9[, manchm8[l7[ |7]7a|8]8ma|9]9 10[mit
eigen|10]10und 11[hatt|11]11besass einen eigenen Hof.|12]20.|21 30{
```

Ex. 4: Dancing turn in S-notation

Source: sf_zvz_061123_2150_stadelmann_bauernsterben_snt

The final text reads:

```
Meinrad Etterlin was a farmer with an own farm. Five years ago, he gave it up, since
then he works as municipial worker for the village Merenschwand.
```

Ex. 5: Final text produced by dancing turn

Source: sf_zvz_061123_2150_stadelmann_bauernsterben_text

The repeated reformulations appear in the progression graph as a dancing turn (see revisions in process 1 – 56 in Fig. 15).

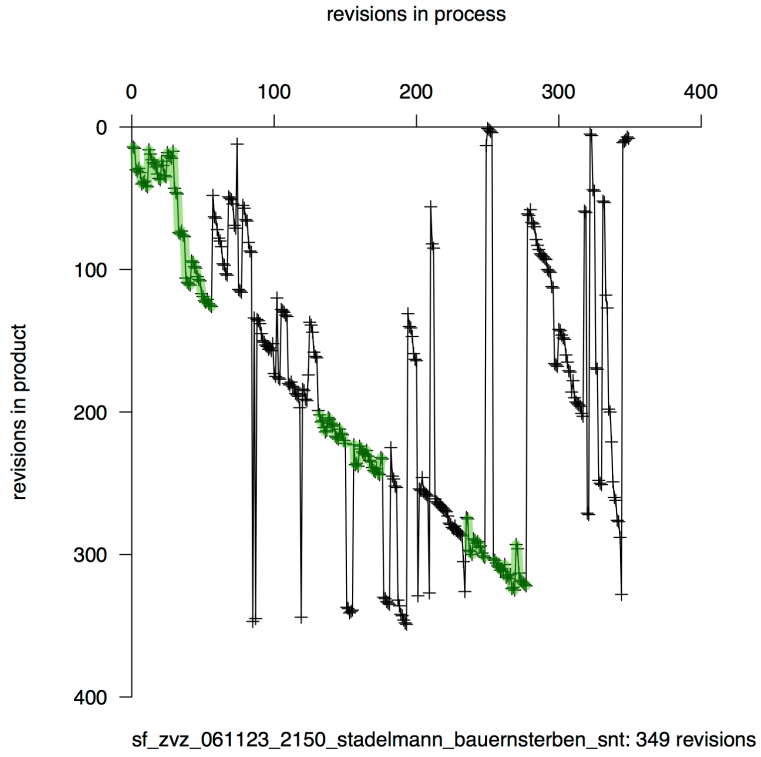

Fig. 15: Initial dancing turn for one paragraph

The screen recordings[37] of this writing process show that S.E. made these 56 revisions within 22 minutes. Considering the message's conceptual low degree of complexity, the question of why S.E. revises this very first paragraph so much arises. In the verbal protocol (see section 3.1.1 for a description of this data type) of another writing process[38], the same author S.E. substantiates and generalizes this revision behavior as follows:

```
580    For me that is such a –
581    I think
582    and I write
583    And then it forms – a little like playdough
```

Ex. 6: Epistemic writing in a dancing turn I
Source: sf_zvz_061123_2150_stadelmann_bauernsterben_verbal

In the same verbal protocol, after being asked about the reason of a revision – a correction of a typo – she answers:

```
894    It has also a little –
895    I realize that now –
896    this rather mechanic aspect of correcting typos
897    leaves you a little space in your mind
898    you can think a little
899    finally you do something
900    that is little bit like holidays for me
```

Ex. 7: Epistemic writing in a dancing turn II
Source: sf_zvz_061123_2150_stadelmann_bauernsterben_verbal

Evidently, in these cases, the revision pattern of dancing fulfills an epistemic function for S.E. While making minor adjustments to the text produced so far, she still has resources for other cognitive activities. In sum, these revisions frame the item piece-by-piece, though what she uses the "little space" in her mind for remains obscure at these positions of the verbal protocol.

4.2.3. Skipping

Skipping turns are dominated by large hops through the text, either moving down or back up. When skipping turns are observed on the screen, the cursor repeatedly skips across longer distances of the text, always in the same direction. Between the skips, single revisions or short sequences of revisions are made.

Skipping: Writing movement in which the writer skips repeatedly over longer distances in the same direction in the text, performing few revisions between the skips.

In the progression graph, skipping draws steep and long lines, either from top to bottom or vice versa, interrupted by a few single revisions or groups of few revisions (see the revisions highlighted in yellow in Fig. 16). In most analyses, skipping also includes very short walking or dancing sequences.

[37] sf_zvz_061123_2150_stadelmann_bauernsterben_desktop
[38] sf_zvz_061128_2150_stadelmann_kabelknatsch_verbal

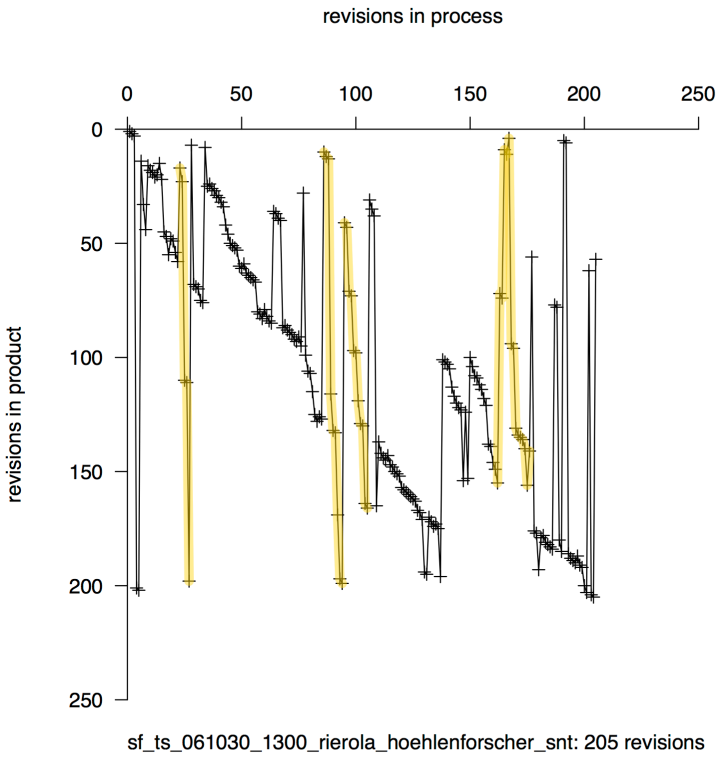

Fig. 16: Phases on the turn level: skipping

Skipping turns are sub-classified either as skipping up or skipping down. A skipping up turn can be directly followed by a skipping down turn, and vice versa.

From a cognitive perspective, the material activity of skipping can be interpreted as planning. Text is being organized, not written down. In skipping turns at the beginning of a progression graph, the writer typically notes down subheadings or keywords representing the text structure. When skipping occurs at the end of a progression graph, it often represents global revising: The writer revises key elements such as subheadings of an almost finished text from top to bottom (or vice versa).

Another revision behavior that results in skipping turns and that can occur after some text is produced, is adding information to or altering the existing information of the text produced so far. In news journalism, recent information from newswires and other sources can prompt the journalist to integrate them in their text. In the progression graph above, the second skipping turn, ranging from revision 86 to 105, is such a case. After having written the first two paragraphs of an item about a speleologist who had an accident in a cave located in the south of Switzerland, the journalist K.R. checks the newswires and reads an updated version of the article.[39] Then, she adds the name of the cave and details about the Swiss-Italian research group in the corresponding parts of the text (see the revisions highlighted in yellow in Ex. 8).

[39] sf_ts_061030_1300_rierola_hoehlenforscher_desktop_2, TC: 00:00 – 01:09

```
97{in der 163{"}163|164Grotta B 98[o|98] 98ossi 164{"}164|165 überhaupt }97|99wieder an die Luft
kommt. 32[Sie |32] 32,33[Kurz darauf |33] 33
187[xx]187|188 188{17}188|189" 60[
57[Kurz darauf]57|58]60|61}29|34 58{61[Die 59[Ret|59]59Teams konnten ]61|62}58|60 62{
63[Gena|63]63Noch 69[ist nicht ]69|70}62|64 68{70[klar]70|71}68|69 71{ha 72[ben|72]72,75[t
73[weder|73]73,74[d|74]74weder|75]75ben die Teams nicht 168[genau 76[er kommuniziert,|76]76er
gesagt]168|169 169{erklärt}169|170, 99{wie und }99|100wo sie den Mann 100{genau
}100|101gefunden haben.}71|77 78{ Klar ist, 150[ 138{ 139[die Hile |139]139für
}138|140de 140[r]140|141]150|151}78|79 151[ 141{n }141|142 79[

]79|80 80{ }80|81]151|152 152{153[er|153]153der }152|154 25{26[nach dem ]26|27 40-
jährig 154[ 142{n}142|143e]154|155 155{e}155|156 81[n]81|82 italienische 89[n]89|90 143{156[n]156|157}143|144
Höhlenforscher |26}25,101{aus der Lombardei }101|102 144{157[kam 145[jede Hilfe|145]145die
Hilfe zu spät. 146{
148{xx"}148|149
}146|147Er |146]157|158}144,82{gehörte zu eine 84[r ]84|85 85{m }85,83[d|83]83,102{4-köpfigen 103[
|103]103 _}102|104 170[schweizerisch-italienischen|84
]170|171 90[Team]90|91}82|86 91{Forscherteam 171{ aus Ital 172[ei|172]172ien und der
Sc 173[shei|173]173hweiz}171|174. 110{158[
]158|159 159{_}159|160 174{176[
```

Ex. 8: Skipping turn in S-notation

Source: sf_zvz_061123_2150_stadelmann_bauernsterben_snt

In the case of television news journalism – and other multimodal forms of journalism such as radio – adding or adjusting meta tags can result in skipping turns as well. As meta tags, placeholders are described that allow the editing system to dub in video bites or inserts[40] at the time specified by the journalist. In the news editing system of *Tagesschau* and *10 vor 10*, these meta tags appear in blue (see Fig. 17) and the time codes are added in the text editor window as well. In the news editing system of *Télejournal* the time codes are only added in the text editor window. If the news item is altered on the time scale, these meta tags have to be adjusted, which often results in a skipping down turn.

[40] Inserts appear as overlay on (TV) screen and contain headlines, names of quoted people, subtitles and other additional information in text form.

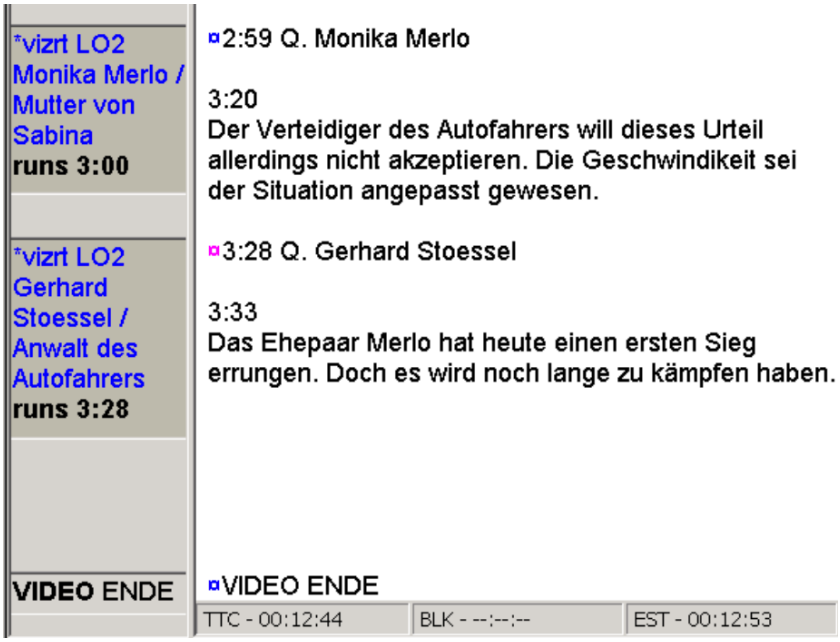

Fig. 17: Meta tags in the editorial system

Source: sf_zvz_061123_2150_stadelmann_bauernsterben_desktop

4.2.4. Jumping

Jumping turns are dominated by large back-and-forth movements. When jumping turns are observed, the cursor position repeatedly alternates between paragraphs, up and down in turn.

Jumping: Writing movement in which the writer repeatedly jumps up and down over longer distances in the text.

In the progression graph of an ideal jumping turn, segments are dominated by spikes. In most analyses, however, the tops of the spikes can consist of small groups of revisions rather than a single revision (see Fig. 18).

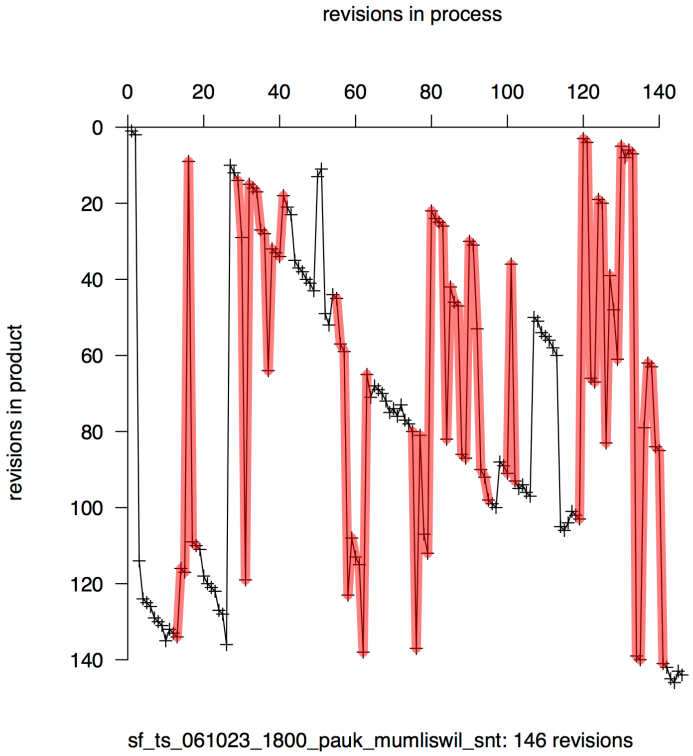

Fig. 18: Phases on the turn level: jumping

From a cognitive perspective, the material activity of *jumping* can be interpreted as revising and reorganizing. Or, text is being moved around rather than being written, which results in a deletion where the text was cut and in an insertion where the text was pasted – to use the terminology of the MID-syntax (see section 3.2.1). Another typical writing activity that results in *jumping* is the parallel revision of two physically distant, but closely related paragraphs, such as the lead and the summary of a text.

Numerous jumping turns can be a sign of a chaotic writing session (see section 4.4.5), but also of a revision behavior that resembles the carving of a statue – with the difference being that parts of the statue are chopped and then attached at another location. One of the few examples of this revision behavior in the corpus can be observed in a writing process of C.P. who writes a news item for *10 vor 10* about a family tragedy that ended with two dead people.[41] She types quickly and a lot and her writing focus switches over the whole text within seconds, often without any observable semantic connection between the different text parts. This results in her longest preliminary version consists of 1355 characters and her final version of 708 characters. This disparity is remarkable because she did not copy and paste outside of her text, except for the introduction, which was written by the anchor.[42]

In the preliminary versions of her text, C.P. differentiates the actual text of her item from other text consisting of notes and preformulated phrases that she occasionally copies and pastes into her actual text by formatting the latter text in red. By doing so, she is able to keep all of the information needed in

[41] sf_ts_061023_1800_pauk_mumliswil_item

[42] These characters were not included in the calculation. In the verbal protocol, C.P. says that she does so, she does not have to switch to another window within the editing system to see how the anchor introduces her item (sf_ts_061025_2400_pauk_wetterfrankreich_verbal, lines 27 – 35). A finely coordinated intertextual connection between introduction and item is an essential feature of news production.

sight and still has a valid estimation of the text length that is automatically calculated by the editing system (see the number at the bottom right of each text version in Fig. 19).

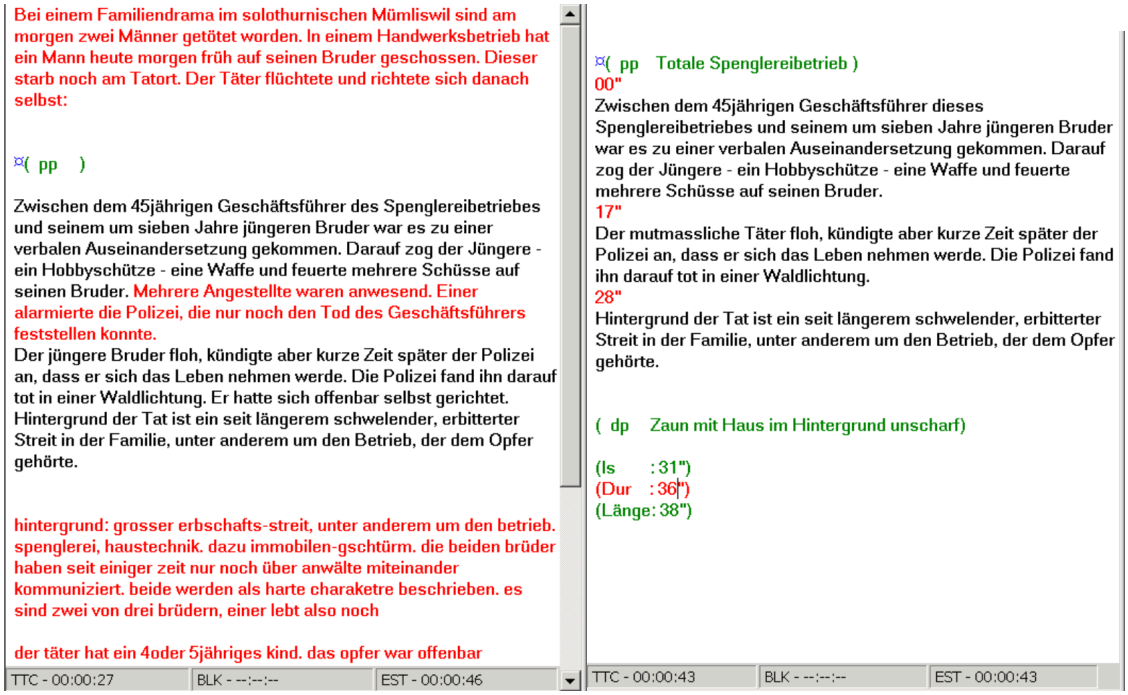

Fig. 19: Preliminary and final version of C.P.'s text

4.2.5. Unclear

Unclear is the residual category between more or less homogeneous revision patterns that can be categorized as one of the four writing turns presented above, there are revisions that do not fit into them and consequently separate them.

4.3. Phases on the run level

On the run level, phases are consistent runs through the entire emerging text. These phases are delimited by discontinuities in the large-scale revision activity throughout the emerging text. In the newsroom, a phase shift on this level can take the dynamic system from drafting an entire news item to revising it, each time from the beginning of the text to end of it.

Writing run: Segment of writing processes that is performed from top to bottom throughout substantial parts of the emerging text.

Two main types of writing phases on the run level can be differentiated, depending on the direction of the movement through the text: The more common writing run in the reading order (down-run) and the more rarely observed writing run from bottom to top of the text produced so far (up-run). Thus, three down-runs can be observed in Fig. 20.

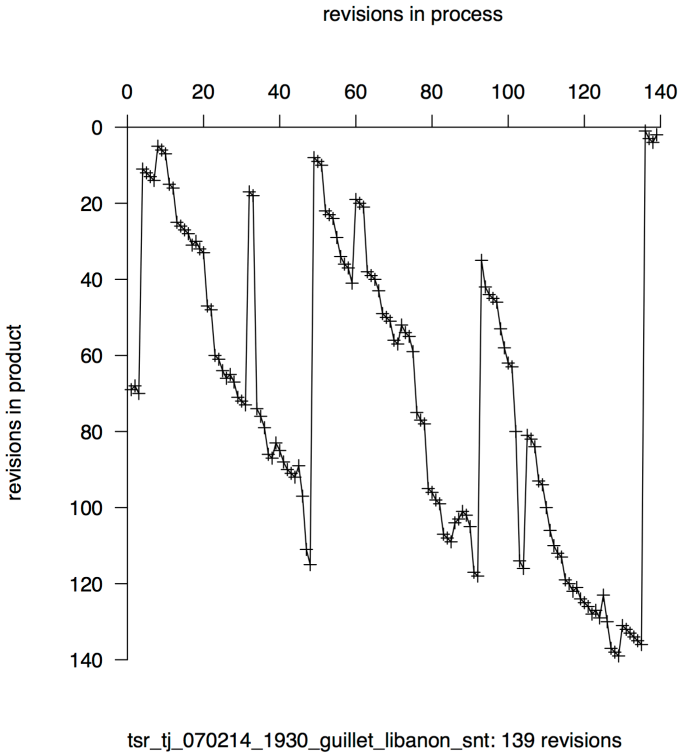

Fig. 20: Exemplification of phases on the run level

On the next higher level of the multilayered phase model, runs are elements of writing sessions. The principle of consistency itself scales up to iterativity, i.e. to various numbers and combinations of runs taking place in an entire writing session. Among other things, these numbers and combinations are the topic of the next section.

4.4. Phases on the session level

On the session level, phases consist around changes in particular settings. These phases are delimited by contextual brakes, respectively by discontinuities in large-scale writing activities, such as a change of workplace. In the newsroom, a phase shift on this level can take the dynamic system of writing from individual drafting at the desk to collaborative video editing in the editing room.

Writing session: Segment of a writing process that is performed in one particular context and is delimited by fundamental contextual changes.

In terms of Dynamic Systems Theory, the writing session is the trajectory of the dynamic system of writing in its state space between fundamental changes of the context.

Such fundamental contextual changes result in new structures, exerting new powers, that enable and constrain the situated activity of writing. Examples of fundamental contextual changes are: changing the workplace (e.g. from desk to site or cutting room), the text file (e.g. after file loss in computer crashes), or the overall activity (from working to leisure). Depending on the context, such events can – but do not have to – fundamentally change the writing context.

Empirically, every writing session of a writing process is usually represented in one single progression graph. However, several writing sessions can and should be combined into one progression graph provided that qualitative information exists that proves that the assumed event of fundamental change

did not fundamentally change the writing context.[43] This happens to be the case if a writer indeed changes the workplace, but seamlessly continues writing where she or he left off before.

On the level of the overall writing process, I use the concept of sessions to differentiate one-session from multi-sessions writing processes. The sessions themselves are distinguished first from a context perspective. Frequent in the newsroom are, for example, workplace sessions, cutting room sessions, and field sessions. From a revision behavior perspective, these sessions are dominated by linear (see section 4.4.1), one-run (4.4.2), multi-run (4.4.3), fragmentary (4.4.4), and chaotic (4.4.5) movements.

4.4.1. Linear session

Linear session: Writing session in which the writer moves from the top of the text to its end in one single walking movement.

When observing linear sessions on screen, the cursor moves once from top to bottom throughout an entire session, with only casual corrections of typographical errors. This happens most often with short writing sessions and texts.

In the progression graph, linear sessions present with a straight line from the upper left to the lower right (see Fig. 21).

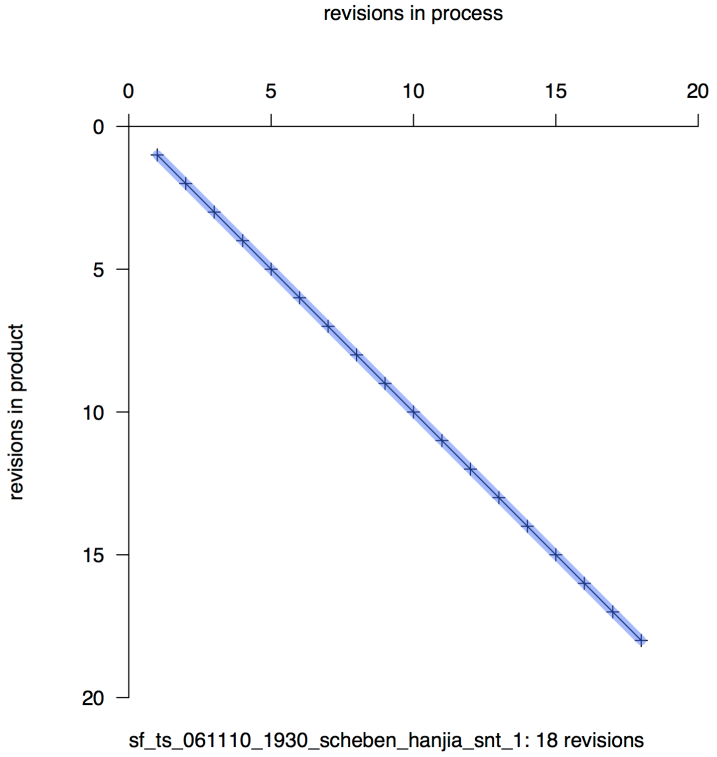

Fig. 21: Phases on the session level: Linear session

In practical analysis, the category of linear session surpasses the prototype case of perfect linearity. Sessions whose progression graphs show minor deviations from linearity, consisting of few revisions

[43] Verbal protocols are the data type of choice to validate if the assumed fundamental change event did effectively fundamentally change the writing context. This is because they contain the writer's explanations for the change. In most cases, however, the screen recordings contain sufficient evidence for the decision whether a session should be combined because they show in detail what the writer did before and after the assumed fundamental change.

(up or down) that interrupt long linear segments of the progression graph, are also classified as linear (see Fig. 22).

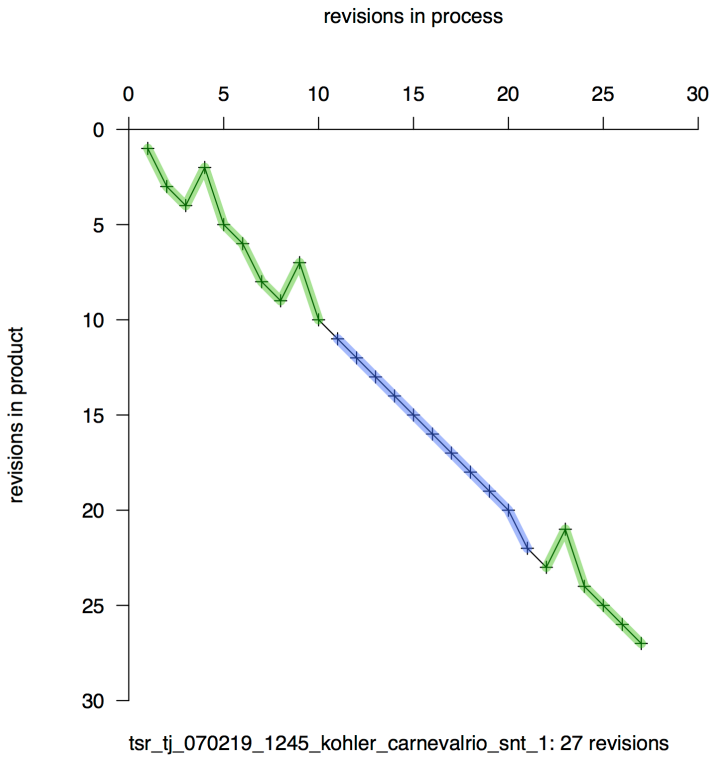

Fig. 22: Linear session with minor deviations

Sessions in which the minor jumps performed in the writing process sum up to an overall impression of oscillation or deviation around the linear movement are not classified as linear. Depending on the application and the data, the degree of linearity can be chosen and defined as the percentage of linear revisions of the writing session. A linear session is a special case of a one-run session, meaning that if a one-run session does not reach the threshold of a linear session it is coded as a one-run session. In addition, a linear session can be one run of multi-run session (see section 4.4.3).

4.4.2. One-run session

One-run session: Writing session in which the writer begins at the top of the text and ends at the bottom, combining walking with dancing movements and some sporadic skips and jumps.

When observing one-run sessions on screen, the cursor moves once from top to bottom throughout an entire session, but the author often stops to revise paragraphs or sentences that have just been written.

In the progression graph, one-run sessions appear as reaching from the upper left to the lower right, but in contrast to the linear session, with fluctuation and outliers (see Fig. 23).

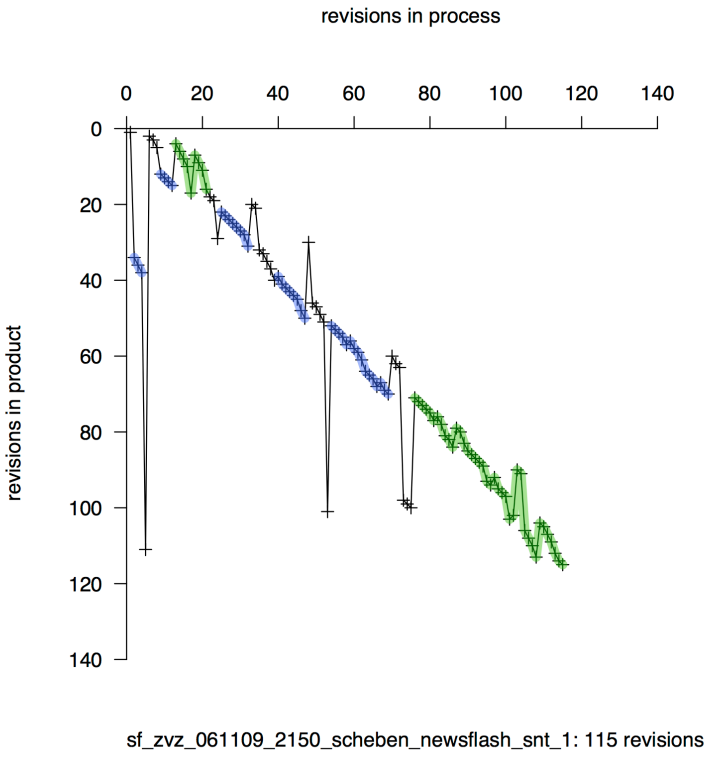

Fig. 23: Phases on the session level: One-run session

In practical analysis, the fluctuations consist of walking, dancing, or jumping turns. They are part of an overall movement from the beginning to the end of the text produced in the session. The outliers are short linear fragments outside the main top-down movement or of single big jumps. Similar to the linear session (see section 4.4.1), the application and the data determine how much deviation from the top-down movement is acceptable.

Sessions where jumps and fragments are not seen as fluctuations and outliers of one single top-down movement are not classified as one-run session, but as multi-run sessions.

4.4.3. Multi-run session

Multi-run session: Writing session in which the writing moves at least twice through the emerging text, combining walking with dancing movements and some sporadic skips and jumps.

When observing multi-run sessions on screen, a first version of the text is developed more or less from top to bottom, and then the author continues to go over it several times.

In the progression graph, multi-run sessions appear as several large top-down movements, mostly with fluctuation and some outliers. This movement results in a sawtooth pattern with at least two spikes (see Fig. 24).

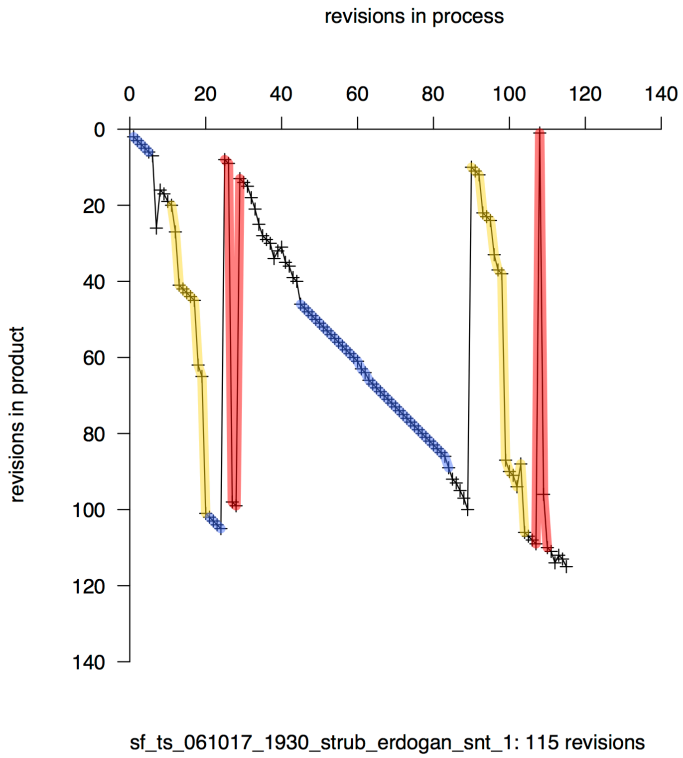

Fig. 24: Phases on the session level: Multi-run session

In practical analysis, new runs (or saw teeth) start near the top of the graph, this is near the beginning of the evolving text. The aforementioned fluctuations consist of walking, dancing, or jumping turns. They are still part of the two or more overall top-down movements. The outliers consist of short linear fragments outside the main top-down movements or of single big jumps. Before, between or after the runs, a multi-run session can contain non-run-like passages such as jumping segments.

Not classified as multi-run are sessions dominated by fragmentary, short top-down movements (see next section).

Subtypes of multi-run sessions:

a) runs 1 then revising

b) runs 2 then jumping

c) runs 2 then the borders

d) runs 2 with fragments

e) runs 3 (revising after 2 runs)

f) runs 3 (walking middle)

g) runs 3 planning-formulating-revising

4.4.4. Fragmentary session

Fragmentary session: Writing session in which the writer jumps or skips up or down between rather short walking and dancing movements.

When observing fragmentary sessions on screen, parts of the text develop in a quite linear way, but the cursor jumps between paragraphs. For example, instead of writing the first paragraph, proceeding to the

second, and ending with the third, the author writes a part of the third paragraph first, then writes the first paragraph, works again on the third, and ends with the second.

In the progression graph, plateaus dominate each fragmentary session. Each plateau is delimited by at least one jump and represents the more or less linear production of a text part (see Fig. 25).

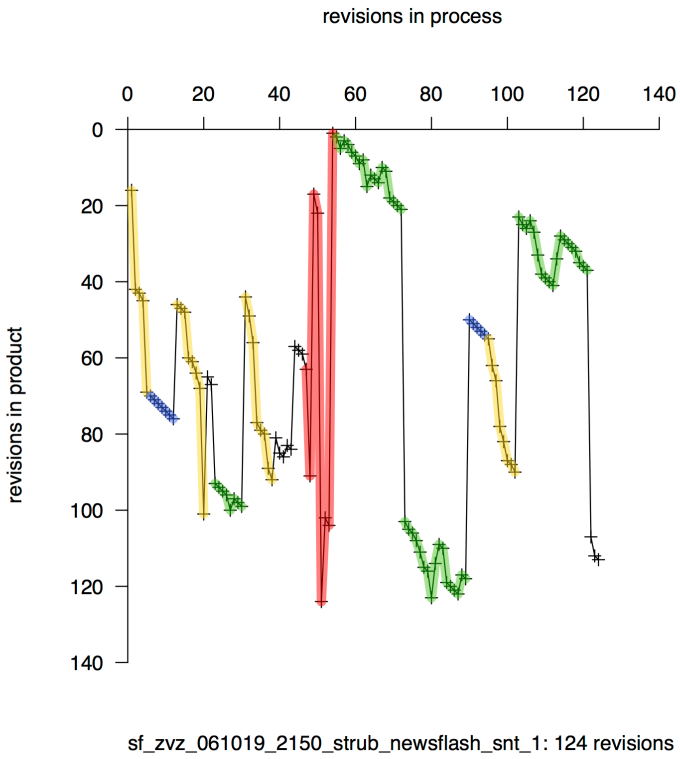

Fig. 25: Phases on the session level: Fragmentary session

In practical analysis, the plateau can be separated by short segments of jumping or skipping turns, or by chaotic behavior. The overall movement, however, is fragmented linear text production.

Sessions where chaotic behavior dominates the overall movement are not classified as fragmentary are (see next section).

4.4.5. Chaotic session

Chaotic session: Writing session dominated by jumping and non-identified movements.

When observing chaotic sessions on screen, the cursor jumps back and forth in the text because the author works oscillatingly on several parts of the text.

In the progression graph, chaotic sessions are dominated by jumping and skipping turns, the absence of longer walking and dancing turns, and the lack of an overall top-down movement.

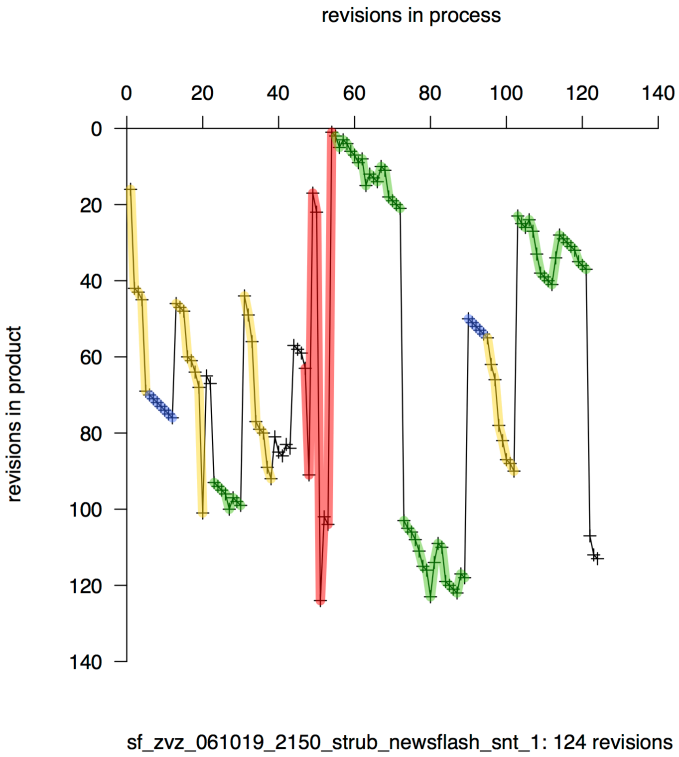

Fig. 26: Phases on the session level: Fragmentary session

In practical analysis, all of the sessions that are not identified as linear, one-run, multi-run, or fragmentary are classified as chaotic. Thus, this represents the residual category on the session level.

4.5. Modeling writing phases on the turn level

For the development of the multilayered phase model presented above, the phases on the turn level are the foundation. With the exception of the phases on the chunk level, all levels build upon the phases on the turn level. Consequently, these phases are the right candidates for the intricate and complex modeling process.

As described in section 3.2.2, the writing phases were initially coded by hand, which is what enabled the creation of numerical features that describe the structure of the data. The creation of these features is an iterative process: First, the researcher analyzes the data properties such as scale of measurement, extrema, averages, jumps and quantiles. In terms of writing processes, the extrema can be the beginning and the end of the writing process or final text, but also the longest jump, i.e. when the writer jumps over the biggest amount of revisions. The creation of features can be inspired by qualitative knowledge about writing processes but it does not necessarily have to. Features simply have the job to describe the data and their value is only determined by how much they contribute to fit a better model, i.e.: Which features and in which combination are suited best to classify several revisions as the same writing phase on the turn level as the linguist did by qualitative coding?

As a second step of the iterative process of feature creation and model building, the researcher combines several single property features into more complex features. Each phase on the turn level is defined by a combination of these features. The combination of the superordinate features (see Fig. 27) and their values result in a total of 60 features (see the R-Code for the creation of all features in section 7.4.2 in the appendix). An example of such a combination and one of the 60 features is the feature

jump.quantiles.back.q0.9.w0.1. This feature checks in a classification window of *10* percent of the writing session if there is a *jump* in *backward* direction over *90* percent of the writing product.

The researcher, as a third step of the iterative process, computes a model with the chosen features and tests with a confusion matrix, explained further below, if the they contribute to fit a better model or not. He then goes back to step one of the iterative process of feature creation and model building, creates new features, combines them differently and computes a new model. He stops the iterative process when its continuation does not improve the model anymore or the model is fitting sufficiently for the intended application.

win.size	Size of the classification window, i.e. the number of revisions that are classified
fuzzy	Number of jumps that are allowed for walking turns
seq.size	Minimal length of a walking turn
nr.dir.change	Number of directional changes (relevant for dancing, skipping and jumping turns)
cut.size	Maximal height of jumps (e.g. for differentiating dancing from jumping turns)
p.jumps	Share of jumps in the classification window that are permitted to be higher than cut.size (e.g. for differentiating dancing from jumping turns)
nr	Vector c(x1, x2) for defining the minimal length of a jumping turn x1 = Minimal number of forward jumps x2 = Minimal number of backward jumps
quantiles	Quantiles to be calculated for the classification of jumps (0.1, 0.3, 0.5, 0.7, 0.9)
longest.move	Six binary features that contain the information whether a jump (10, 20, 35) is in the classification window and if yes, forward or backward

Fig. 27: Features for the random forests model

All features exhibit binary decisions for the decision trees that cumulatively aggregate to random forests. As described in section 3.2.2, the numerous random forests vote for a classification and the majority defines to which writing phase on the turn level the group of revisions within the size of classification window belongs to.

It depends on the creation of the features if the random forests model is able to classify the writing phases correctly. To calculate the validity of the model, a confusion matrix is calculated. The confusion matrix simply shows how many revisions were correctly classified by the model – by comparing with the qualitative coding. If the confusion matrix classifies not precise enough other features can be extracted that enable a better classification.

On the top right of Fig. 28 the confusion matrix of one writing process is exhibited. It reads as follow: Whereas the classification by qualitative coding is situated in the columns, the classification by the model is found in the rows. The classification is also visually recognizable, as the revisions classified

by qualitative coding are marked with a circle and the model's classification by a square. The overall misclassification rate of this writing session amounts to 32 percent and is mainly caused by the misclassification of walking and dancing turns.

To overall misclassification rate of all 16'847 revisions is 28 percent. This value appears to be high, but in statistical terms it is acceptable since the successful modeling for itself is already an achievement.

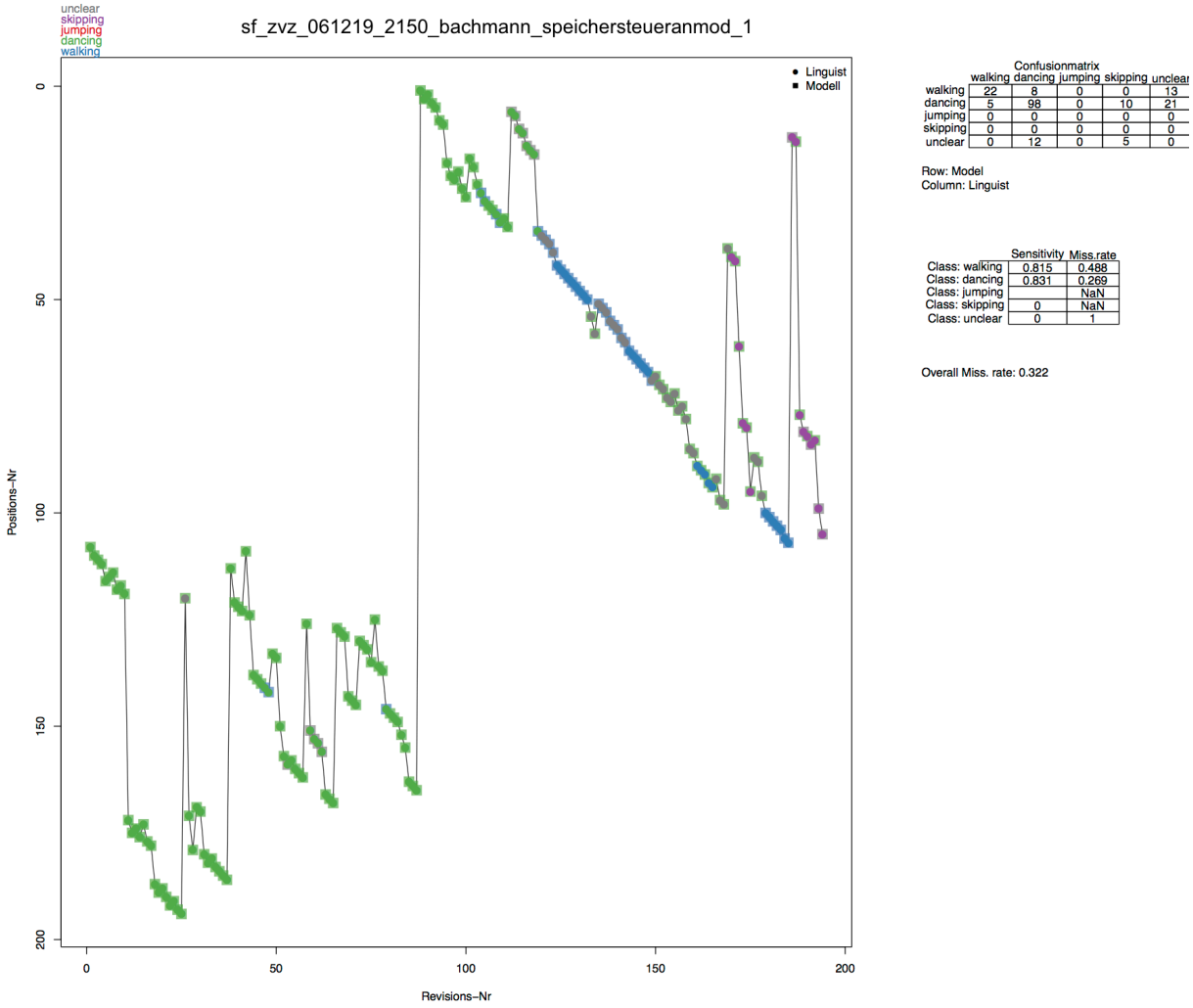

Fig. 28: Exemplification of a confusion matrix

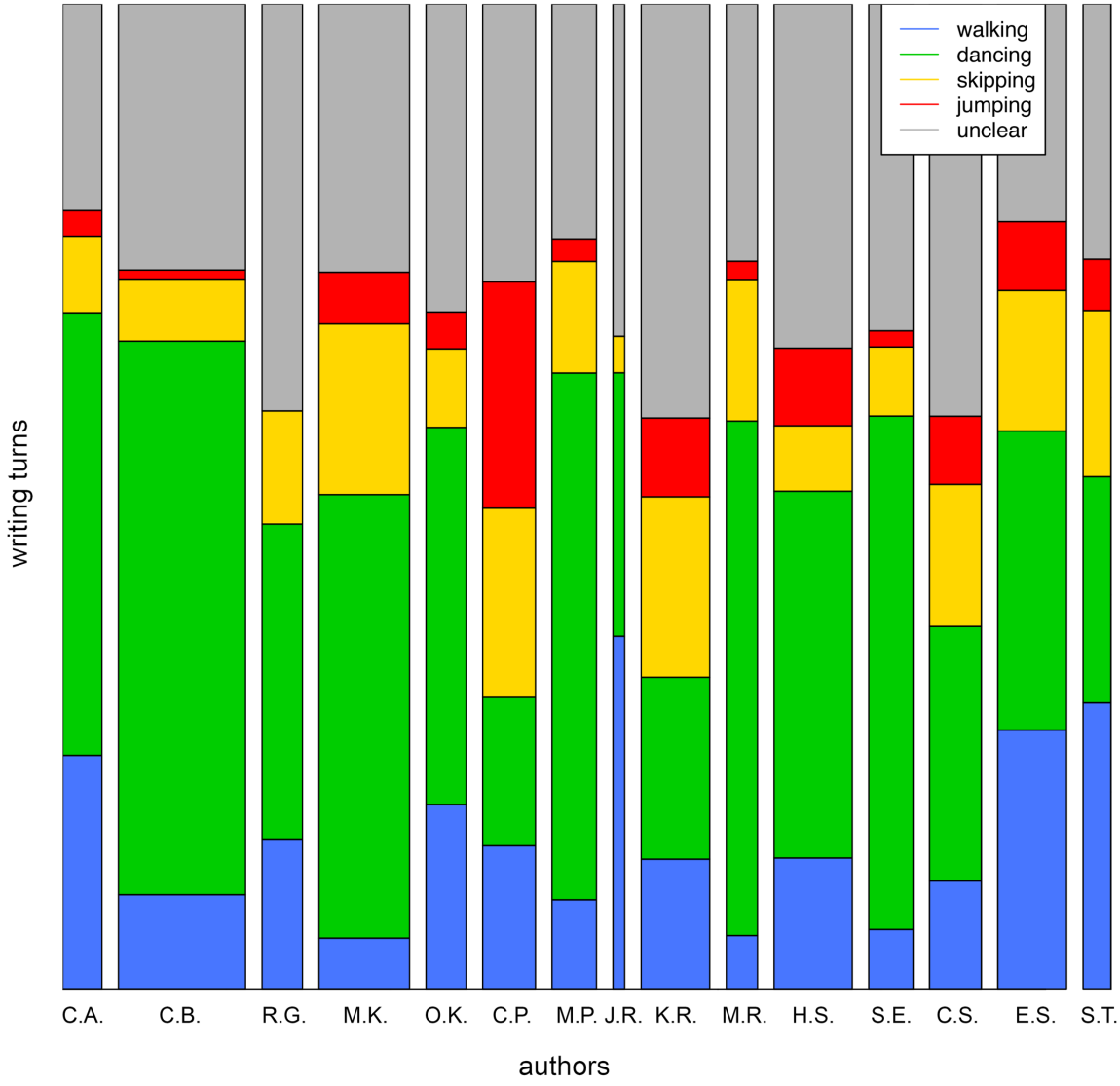

Fig. 29: Distribution of writing turns by authors

Fig. 29 shows the distribution of writing turns by authors. The width of the columns signifies the amount of revisions that were recorded be the particular author.

5. Interpretation

In the second part of his article from 2012, Hayes models children's writing processes with running programs drawing on Fuller (1995). He writes three running programs in the Python language that are able to produce short texts that reflect the complexity of children's writing. To do so, he builds a data base with over 80 statements that he hierarchically assigns to three topic levels, so that the running program can deepen a topic. If the running program choses the sentence "She has a new computer," for example, it is able to deepen the topic to a sentence such as "The computer was a Christmas present." One of the three running programs, the topic-elaboration model, is shown in Fig. 30).

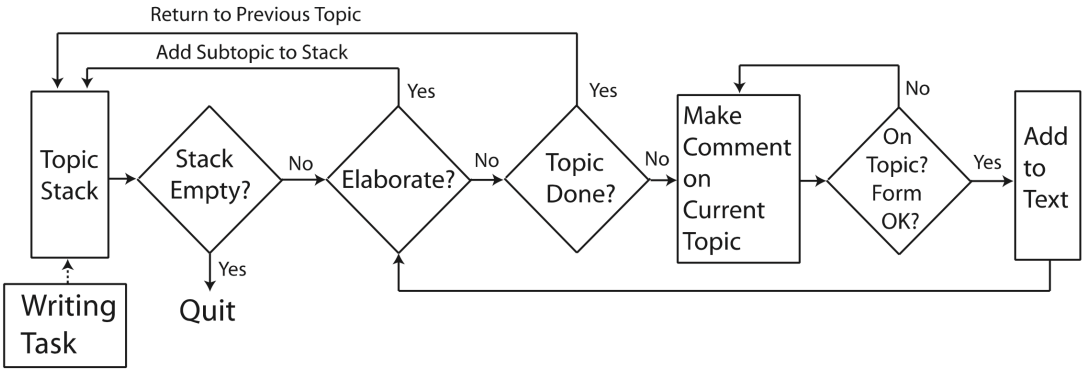

Fig. 30: Hayes' (2012) topic-elaboration model

In his conclusion, Hayes (2012) emphasizes the strengths of the modeling approach:

The expansion of my interest to include children's writing has led me to propose some elaborations of Bereiter and Scardamalia's knowledge-telling model. In turn, modeling children's writing, which is in some ways much simpler than adult writing, has given me courage to try modeling with running programs (a task that seemed once and, perhaps, seems still, too difficult to accomplished with adult writing). I believe that using running programs to model writing is fundamentally superior to using box-and-arrow models. Running programs force us to be very specific about how writing processes work and about the structure of the memory resources that the writing processes rely on. (pp. 385-386)

Although Hayes' approach to modeling writing differs from mine, I share his opinion that modeling – be it with running programs or machine learning methods – forces scholars of writing processes to be concise when they describe revision patterns. Although machine learning methods are more flexible than running programs, the features described in section 4.5 had to be specified concisely. Without the qualitative analysis and the iterative, recursive, and abductive coding described in section 3.1 this would not seem feasible to me.

Generally, I assess the multilayered phase model presented in this book as a substantial contribution to the investigation of writing processes in general, and writing phases in particular. The heretofore unseen application of machine learning methods on writing process data appears very fruitful, especially considering the more than exponential development in computer technology. Naturally, it takes a lot of coding work to extract features of a dataset and build a random forests model, but once this work is done, large corpora of writing process data can be classified within seconds.

The presented modeling approach does, however, have its limitations. First, the complexity of such models may turn them into a sort of black box for people who are not acquainted with them. Although this seems like a general problem of a society that increasingly outsources complex decision making

processes to machines and algorithms, researchers, especially computer scientists but also linguists who use those models, should contribute to "demystify[ing] the black box for non-experts by creating algorithms that can inform, collaborate with, compete with, and understand users in real-world settings" (Boyd-Graber, 2016, p. XXIII).

Another criticism that may arise is that the random forests model classifies 30.5 percent of the revisions as "unclear" (see also Fig. 29 in section 4.5). However, this classification is only the result of not having qualitatively classified them beforehand. In this matter, it is important to record acknowledge that it was never the aim of the model to equip every revision with meaning. Instead, this "fuzziness" is desired and may even be required in order to find the model's appropriate level of abstraction. Otherwise, the model would not fulfill its function of reducing the complexity of such a complex iterative pattern as found in a writing process consisting of hundreds of revisions.

As shown in section 4.5, the random forests model was internally validated by letting it predict the writing turns of an author without including this author's writing processes. Thus, it was trained with all other writing processes. It would be interesting to see how the model classifies new writing process data. For this external validation, the new data has to be transformed into the comma-separated value format and a sample of it has to be coded manually. For similar writing process data, the external validation can be expected to be sufficient, but it would also be productive to try an external validation with distinct writing process data – and then to compare these quantitative results with the qualitative insights again.

The analysis of the combination and sequence of writing turns on the session level (see section 4.4) reveals planning and implementation issues. It also shows the empirically grounded explanation for why some combinations of writing turns are more goal leading in process terms than others. Due to the lack of classification of the writing products, however, the described revision patterns on the session level can not be related to the quality of the text. Hence, an undesirable input from the process-perspective, such as chaotic writing sessions, can lead to desirable outcomes from the product perspective in the end. In short, unorganized writing may result in "good" texts.

This phenomenon is what Elbow refers to when he invites his readers to "just let go" during the, what he frames the first, more creative step of writing (see section 2.2). On the other hand, Elbow (1998a) also supports a more systematic approach when it comes to the second, more critical step of writing:

> But you don't have to give in to this dilemma of creativity versus critical thinking and submit to the dominance of one muscle and lose the benefits of the other. If you separate the writing process into two stages, you can exploit these opposing muscles one at a time: first be loose and accepting as you do fast early writing; then be critically toughminded as you revise what you have produced. What you'll discover is that these two skills used alternately don't undermine each other at all, they enhance each other. (p. 9)

Yet, for professional writing, and especially for television news journalism, a controlled writing process, i.e. one that is finished before the deadline is more favorable than a uncontrolled writing process that may result in an exceptional news item, but can't because the deadline is passed and the item cannot be broadcasted.

6. Conclusion and outlook

This book presented a multilayered model of writing phases – defined as temporal segments of writing processes that are dominated by a particular writing activity. Drawing on a corpus of 120 multimodal writing processes of Swiss television journalists, four scales of writing phases have been identified and one scale has been modeled by machine learning methods. Since the data was originally collected under an ethnographic research framework, the results of the statistical modeling were related to contextual conditions such as the writing environment, the writing task, and the experience of the writers.

As introduced in section 5, a desideratum for future research is relating the reported process-oriented results to product-oriented results. Further, machine learning methods are well suited for this endeavor – even more so because they are much more advanced for applications in text linguistics than they are for writing process research.

As with every new endeavor, it is best to first demarcate the territory, and this project has done just that. Now that territory has been demarcated, future research can aim toward providing a detailed analysis of the landscape.

7. Appendix

7.1. List of figures

7.2. List of excerpts

7.3. Questionnaire for the review protocol

The review protocol is a semi-structured interview that was lead directly after the journalist under investigation had finished the text production process.

a) What is the item about?

b) Which message do you want to get across with the item ?

c) How did you proceed with the production?

d) Which guidelines did you have for this news item?

e) Who issued the guidelines?

f) Did you talk with colleagues about the item? At which stage of the process?

g) Who did proof read?

h) On which level were the feedbacks?

i) Which preliminary products did you use?

7.4. R-Scripts

7.4.1. Feature creation

```
1   ### ////////////////////////////////////////////////
2   ### modeling writing phases
3   ### :: data preparation > create all features
4   ### update: feb 13
5   ### Author: Beate Sick (ZHAW)
6   ### ////////////////////////////////////////////////
7
8
9   rm(list = ls())
10
11  ### ------------------------------------------------------------
12  ### load packages
13  ### ------------------------------------------------------------
14
15  packages <- c("plotrix",
16                "randomForest",
17                "gplots",
18                "ggplot2",
19                "caret",
20                "cluster",
21                "colorspace",
22                "class",
23                "splines",
24                "som",
25                "Hmisc"
26                )
27
28  whichNotInstalled <- !(packages %in% installed.packages()[,"Package"])
29
30  if(any(whichNotInstalled))
31    install.packages(pkgs = packages[whichNotInstalled], quiet = TRUE)
32
33  sapply(X = packages, FUN = library, character.only = TRUE, quietly = TRUE)
34
35  rm(list = c("packages", "whichNotInstalled"))
36
37  ### ------------------------------------------------------------
38  ### read data
```

```
39      ### ------------------------------------------------------------
40
41      data_all.path <- "data/all/data_all.RData"
42
43      load(data_all.path)
44      dat <- mat
45
46      ### ------------------------------------------------------------
47      ### Source functions for features creation
48      ### ------------------------------------------------------------
49      source("functions_for_feature_creation.r")
50
51      ### ------------------------------------------------------------
52      ### definitions
53      ### ------------------------------------------------------------
54
55      position <- dat$position_nr
56      unit <- dat$text
57
58      ### ------------------------------------------------------------
59      ### new wrapper function by2
60      ### ------------------------------------------------------------
61
62      by2 <- function(data = position, INDICES = unit, FUN = walking, ...)
63      {
64        y <- by(data = data, INDICES = INDICES, FUN = FUN, ...)
65        unlist(y[unique(unit)])
66      }
67
68      ### ------------------------------------------------------------
69      ### issue with infinite values
70      ### ------------------------------------------------------------
71      ### if data sets are too small, or missing values, then 'INF' values are generated.
72      ### First, proof, what causes them (e.g. too small data set), and second, set them
        to NA
73
74      ### ------------------------------------------------------------
75      ### create features: walking
76      ### ------------------------------------------------------------
77
78      ### win.size: Will be subtracted by one if the number is odd
79      ### fuzzy:    Maximum number of jumped revisions, 0=strict walking
80
```

```
81      # strict walking: measures how long the sequence of strict-walking is in the window
82      my.par <- c("w4.f0")
83      x <- by2(data = position, INDICES = unit, FUN = walking, win.size=4, fuzzy=0)
84      dat <- cbind(dat, walking = factor(x))
85
86      my.par <- c(my.par,"w10.f0")
87      x <- by2(position, unit, walking, win.size=10, fuzzy=0)
88      dat <- cbind(dat, walking = factor(x))
89
90      my.par <- c(my.par,"w20.f0")
91      x <- by2(position, unit, walking, win.size=20, fuzzy=0)
92      dat <- cbind(dat, walking = factor(x))
93
94      my.par <- c(my.par,"w10.f3")
95      x <- by2(position, unit, walking, win.size=10, fuzzy=3)
96      dat <- cbind(dat, walking = factor(x))
97
98      my.par <- c(my.par,"w15.f5")
99      x <- by2(position, unit, walking, win.size=15, fuzzy=5)
100     dat <- cbind(dat, walking = factor(x))
101
102     my.par <- c(my.par,"w20.f5")
103     x <- by2(position, unit, walking, win.size=20, fuzzy=5)
104     dat <- cbind(dat, walking = factor(x))
105
106     my.par <- c(my.par,"w25.f7")
107     x <- by2(position, unit, walking, win.size=25, fuzzy=7)
108     dat <- cbind(dat, walking = factor(x))
109
110     ### ------------------------------------------------------------
111     ### create features: nr. of direction changes
112     ### ------------------------------------------------------------
113     ### win.size: Will be subtracted by one if the number is odd
114     ### is.na(x) <- is.infinite(x)
115
116     my.par <- c(my.par,"w5")
117     x <- by2(position, unit, nr.dir.change, win.size=5)
118     x[is.infinite(x)] <- 0
119     dat <- cbind(dat, nr.dir.changes = x)
120
121     my.par <- c(my.par,"w10")
122     x <- by2(position, unit, nr.dir.change, win.size=10)
123     x[is.infinite(x)] <- 0
```

```
124       dat <- cbind(dat, nr.dir.changes = x)
125
126       my.par <- c(my.par,"w20")
127       x <- by2(position, unit, nr.dir.change, win.size=20)
128       x[is.infinite(x)] <- 0
129       dat <- cbind(dat, nr.dir.changes = x)
130
131
132       ### ------------------------------------------------------------
133       ### create features: nr. of walking sequences
134       ### ------------------------------------------------------------
135
136       ### win.size: Will be subtracted by one if the number is odd
137       ### seq.size: length of the walking turn
138
139
140       my.par <- c(my.par,"w10.s3")
141       x <- by2(position, unit, nr.walking.seq, win.size=10, seq.size=3)
142       x[is.infinite(x)] <- 0
143       dat <- cbind(dat, nr.walk.seq = x)
144
145       my.par <- c(my.par,"w15.s4")
146       x <- by2(position, unit, nr.walking.seq, win.size=15, seq.size=4)
147       x[is.infinite(x)] <- 0
148       dat <- cbind(dat, nr.walk.seq = x)
149
150       my.par <- c(my.par,"w25.s6")
151       x <- by2(position, unit, nr.walking.seq, win.size=25, seq.size=6)
152       x[is.infinite(x)] <- 0
153       dat <- cbind(dat, nr.walk.seq = x)
154
155       ### ------------------------------------------------------------
156       ### create features: strict dancing indicator
157       ### ------------------------------------------------------------
158       ### win.size: Will be subtracted by one if the number is odd
159       ## cut.size: Maxium permitted jump size
160       ## p.jumps:  share of jumps in the classification window that are permitted to be
          higher than cut.size
161
162       # y <- by(position, unit, strict.dancing, 10, 7, 0.25)
163       # x <- unlist(y[unique(unit)])
164       # dat <- cbind(dat, strict.dancing.ind=x)
165
```

```
166       my.par <- c(my.par,"w10.cs7.pj0.1")
167       x <- by2(position, unit, strict.dancing,
168               win.size=10, cut.size=7, p.jumps=0.1)
169       dat <- cbind(dat, strict.dancing.ind=factor(x))
170
171       my.par <- c(my.par,"w20.cs8.pj0.2")
172       x <- by2(position, unit, strict.dancing,
173               win.size=20, cut.size=8, p.jumps=0.2)
174       dat <- cbind(dat, strict.dancing.ind=factor(x))
175
176       my.par <- c(my.par,"w25.cs5.pj0.25")
177       x <- by2(position, unit, strict.dancing,
178               win.size=25, cut.size=5, p.jumps=0.25)
179       dat <- cbind(dat, strict.dancing.ind=factor(x))
180
181       my.par <- c(my.par,"w25.cs8.pj0.25")
182       x <- by2(position, unit, strict.dancing,
183               win.size=25, cut.size=8, p.jumps=0.25)
184       dat <- cbind(dat, strict.dancing.ind=factor(x))
185
186       ### ------------------------------------------------------------
187       ### create features: fuzzy walking indicator
188       ### ------------------------------------------------------------
189       ### win.size: Will be subtracted by one if the number is odd
190       ## cut.size: maximum permitted jump size
191
192       my.par=c(my.par,"w6.cs2")
193       x <- by2(position, unit, fuzzy.walking, win.size=6, cut.size=2)
194       dat <- cbind(dat, fuzzy.walking.ind=factor(x))
195
196       my.par=c(my.par,"w10.cs3")
197       x <- by2(position, unit, fuzzy.walking, win.size=10, cut.size=3)
198       dat <- cbind(dat, fuzzy.walking.ind=factor(x))
199
200       my.par=c(my.par,"w15.cs5")
201       x <- by2(position, unit, fuzzy.walking, win.size=15, cut.size=5)
202       dat <- cbind(dat, fuzzy.walking.ind=factor(x))
203
204       my.par=c(my.par,"w25.cs10")
205       x <- by2(position, unit, fuzzy.walking, win.size=25, cut.size=10)
206       dat <- cbind(dat, fuzzy.walking.ind=factor(x))
207
208       ### ------------------------------------------------------------
```

```
209     ### create features: strict jumping indicator
210     ### ---------------------------------------------------------
211     ## win.size: absolute width of the classification window
212     ## cut.size: minimal jump size, absolute or in percent
213     ## nr:       Vector c(x1, x2)
214     ##           x1 = minimum of demanded forward jumps
215     ##           x2 = maximum of demanded forward jumps
216
217
218     my.par=c(my.par,"w15.cs15.f2.b2")
219     x <- by2(position, unit, strict.jumping,
220               win.size=15, cut.size=15, nr=c(2,2))
221     dat <- cbind(dat, strict.jumping.ind=factor(x))
222
223     my.par=c(my.par,"w25.cs20.f3.b3")
224     x <- by2(position, unit, strict.jumping,
225               win.size=25, cut.size=20, nr=c(3,3))
226     dat <- cbind(dat, strict.jumping.ind=factor(x))
227
228     my.par=c(my.par,"w35.cs30.f3.b3")
229     x <- by2(position, unit, strict.jumping,
230               win.size=35, cut.size=30, nr=c(3,3))
231     dat <- cbind(dat, strict.jumping.ind=factor(x))
232
233     # relative jump size
234     my.par=c(my.par,"w15.cs0.2.f2.b2")
235     x <- by2(position, unit, strict.jumping,
236               win.size=15, cut.size=0.2, nr=c(2,2))
237     dat <- cbind(dat, strict.jumping.ind=factor(x))
238
239     my.par=c(my.par,"w25.cs0.25.f2.b2")
240     x <- by2(position, unit, strict.jumping,
241               win.size=25, cut.size=0.25, nr=c(2,2))
242     dat <- cbind(dat, strict.jumping.ind=factor(x))
243
244     my.par=c(my.par,"w25.cs0.2.f3.b3")
245     x <- by2(position, unit, strict.jumping,
246               win.size=25, cut.size=0.2, nr=c(3,3))
247     dat <- cbind(dat, strict.jumping.ind=factor(x))
248
249
250     ### ---------------------------------------------------------
251     ### create features: strict skipping indicator
```

```
252     ### ------------------------------------------------------------
253     ## win.size: absolute width of classification window
254     ## cut.size: minimal jumping sitze, absolute or in percent
255     ## nr:       Vector c(x1, x2)
256     ##           x1 = minimum of demanded forward jumps
257     ##           x2 = maximum of demanded forward jumps
258
259     # my.par=c(my.par,"w25.cs0.2.f3.b3")
260     #
261     # y <- by(position, unit, strict.jumping, 10, 30, c(2,1))
262     # x <- unlist(y[unique(unit)])
263     # dat <- cbind(dat, strict.skipping.ind=x)
264
265     ### ------------------------------------------------------------
266     ### create features: jump quantiles
267     ### ------------------------------------------------------------
268     ### jump quantiles
269     ## win.size:  width of classification window in percent or absolute
270     ## quantiles: quantiles to be calculated p = c(0.1, 0.3, 0.5, 0.7, 0.9)
271     my.par <- c(my.par,rep("w15.q",5))
272     y <- by(position, unit, jump.quantiles, 15)
273     x <- do.call("rbind", lapply(y[unique(unit)], "["))
274     is.na(x) <- is.infinite(x)
275     #test=cbind(dat, jump.quantiles.back=x)
276     dat <- cbind(dat, jump.quantiles.back=x)
277
278     my.par=c(my.par,rep("w25.q",5))
279     y <- by(position, unit, jump.quantiles, 25)
280     x <- do.call("rbind", lapply(y[unique(unit)], "["))
281     is.na(x) <- is.infinite(x)
282     dat <- cbind(dat, jump.quantiles.back=x)
283
284     my.par=c(my.par,rep("w0.1.q",5))
285     y <- by(position, unit, jump.quantiles, win.size=0.1)
286     x <- do.call("rbind", lapply(y[unique(unit)], "["))
287     is.na(x) <- is.infinite(x)
288     dat <- cbind(dat, jump.quantiles.back=x)
289
290     my.par=c(my.par,rep("w0.2.q",5))
291     y <- by(position, unit, jump.quantiles, win.size=0.2)
292     x <- do.call("rbind", lapply(y[unique(unit)], "["))
293     is.na(x) <- is.infinite(x)
294     dat <- cbind(dat, jump.quantiles.back=x)
```

```
295
296       ### ------------------------------------------------------------
297       ### create features: abs jump quantiles
298       ### ------------------------------------------------------------
299       ### abs jump quantiles
300       ## win.size:  width of classification window in % or absolute
301       ## quantiles: quantiles to be calculated p = c(0.1, 0.3, 0.5, 0.7, 0.9)
302
303       my.par=c(my.par,rep("w15.aq",5))
304       y <- by(position, unit, abs.jump.quantiles, 15)
305       x <- do.call("rbind", lapply(y[unique(unit)], "["))
306       is.na(x) <- is.infinite(x)
307       dat <- cbind(dat, abs.jump.quantiles.back=x)
308
309       my.par=c(my.par,rep("w25.aq",5))
310       y <- by(position, unit, abs.jump.quantiles, 25)
311       x <- do.call("rbind", lapply(y[unique(unit)], "["))
312       is.na(x) <- is.infinite(x)
313       dat <- cbind(dat, abs.jump.quantiles.back=x)
314
315       my.par=c(my.par,rep("w0.1.aq",5))
316       y <- by(position, unit, abs.jump.quantiles, 0.1)
317       x <- do.call("rbind", lapply(y[unique(unit)], "["))
318       is.na(x) <- is.infinite(x)
319       dat <- cbind(dat, abs.jump.quantiles.back=x)
320
321       my.par=c(my.par,rep("w0.2.aq",5))
322       y <- by(position, unit, abs.jump.quantiles, 0.2)
323       x <- do.call("rbind", lapply(y[unique(unit)], "["))
324       is.na(x) <- is.infinite(x)
325       dat <- cbind(dat, abs.jump.quantiles.back=x)
326
327       ### ------------------------------------------------------------
328       ### create features: longest move
329       ### ------------------------------------------------------------
330       ### longest move
331       ## win.size: absolute width of classification window
332       ## pos (logical): if TRUE forward moves are encountered
333       ## pos (logical): if FALSE backward moves are encountered
334
335       my.par=c(my.par,"w35.move.f")
336       y <- by(position, unit, longest.move, 35, TRUE)
337       x <- unlist(y[unique(unit)])
```

```
338       is.na(x) <- is.infinite(x)
339       dat <- cbind(dat, longest.move.pos=x)
340
341       my.par=c(my.par,"w10.move.f")
342       y <- by(position, unit, longest.move, 10, TRUE)
343       x <- unlist(y[unique(unit)])
344       is.na(x) <- is.infinite(x)
345       dat <- cbind(dat, longest.move.pos=x)
346
347       my.par=c(my.par,"w20.move.f")
348       y <- by(position, unit, longest.move, 20, TRUE)
349       x <- unlist(y[unique(unit)])
350       is.na(x) <- is.infinite(x)
351       dat <- cbind(dat, longest.move.pos=x)
352
353       # backward
354       my.par=c(my.par,"w35.move.b")
355       y <- by(position, unit, longest.move, 35, FALSE)
356       x <- unlist(y[unique(unit)])
357       is.na(x) <- is.infinite(x)
358       dat <- cbind(dat, longest.move.pos=x)
359
360       my.par=c(my.par,"w10.move.b")
361       y <- by(position, unit, longest.move, 10, FALSE)
362       x <- unlist(y[unique(unit)])
363       is.na(x) <- is.infinite(x)
364       dat <- cbind(dat, longest.move.pos=x)
365
366       my.par=c(my.par,"w20.move.b")
367       y <- by(position, unit, longest.move, 20, FALSE)
368       x <- unlist(y[unique(unit)])
369       is.na(x) <- is.infinite(x)
370       dat <- cbind(dat, longest.move.pos=x)
371
372       colnames(dat)
373       fcol <- colnames(dat)[-(1:9)]
374       length(fcol)
375       length(my.par)
376       spec.names <- paste(fcol, my.par, sep=".")
377       length(spec.names)
378       head(spec.names)
379
380       colnames(dat)[-(1:9)] <- spec.names
```

```
381
382       ## ------------------------------
383       ## save data
384       ## ------------------------------
385       data.feature.save.path <- "data/feature"
386       write.csv(dat, file = paste(data.feature.save.path, "/data_with_feature.csv",
387                                   sep = ""),
388                 row.names = FALSE)
389       save(dat, file = paste(data.feature.save.path, "/data_with_feature.RData",
390                              sep = ""))
```

7.4.2. Functions for feature creation

```
391       ### functions for extracting local features from profile
392       ### ------------------------------------------------------------
393       # walking ist strict linear (+1). Fuzzy walking means the deviation of strict walking
          (+n).
394
395       # Value: 1. backward, 2. forward, 3. symmetrical,
396       # 4. is true if 1. is true
397
398       walking <- function(x, win.size, fuzzy = 0)
399       ### win.size: will be substracted by one if the number is odd
400       ### fuzzy:    maximum number of jumped revisions, 0=strict walking
401
402       {
403           n <- length(x)
404           ret <- matrix(NA, n, 4)
405           ind <- 1:n
406
407           for (i in 1:n)
408           {
409               ind.back <- (i-win.size+1):i
410               ind.forw <- i:(i+win.size-1)
411
412               if (win.size %% 2) {
413                   ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
414               } else {
415                   ind.both <- (i-win.size/2):(i+win.size/2) }
416
417               if (all(ind.back %in% ind))
418                   ret[i,1] <- sum(!(diff(x[ind.back])==1)) <= fuzzy
419
```

```
420             if (all(ind.forw %in% ind))
421                 ret[i,2] <- sum(!(diff(x[ind.forw])==1)) <= fuzzy
422 
423             if (all(ind.both %in% ind))
424                 ret[i,3] <- sum(!(diff(x[ind.both])==1)) <= fuzzy
425 
426         }
427         return(apply(ret, 1, any, na.rm=TRUE))
428 
429     }
430 
431 
432 
433     ### Number of directional changes
434     nr.dir.change <- function(x, win.size)
435     ### win.size: Will be substracted by one if the number is odd
436     {
437         n <- length(x)
438         ret <- matrix(NA, n, 3)
439         ind <- 1:n
440 
441         for (i in 1:n)
442         {
443             ind.back <- (i-win.size+1):i
444             ind.forw <- i:(i+win.size-1)
445 
446             if (win.size %% 2) {
447                 ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
448             } else {
449                 ind.both <- (i-win.size/2):(i+win.size/2) }
450 
451             if (all(ind.back %in% ind))
452                 ret[i,1] <- sum(abs(diff(diff(x)[ind.back] < 0)))
453 
454             if (all(ind.forw %in% ind))
455                 ret[i,2] <- sum(abs(diff(diff(x)[ind.forw] < 0)))
456 
457             if (all(ind.both %in% ind))
458                 ret[i,3] <- sum(abs(diff(diff(x)[ind.both] < 0)))
459         }
460         return(apply(ret, 1, max, na.rm=TRUE))
461     }
462 
```

```
463
464     ### Number of strict-walking sequences
465
466     # x <- position[author == "revoin"]
467     # win.size=15
468     # seq.size = 4
469
470     nr.walking.seq <- function(x, win.size, seq.size)
471     ### win.size: Will be substracted by one if the number is odd
472     {
473
474         if (seq.size < 3)
475             stop("seq.size must be greater than 2!")
476
477         n <- length(x)
478         ret <- matrix(NA, n, 3)
479         ind <- 1:n
480
481         for (i in 1:n)
482         {
483             ind.back <- (i-win.size+1):i
484             ind.forw <- i:(i+win.size-1)
485
486             if (win.size %% 2) {
487                 ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
488             } else {
489                 ind.both <- (i-win.size/2):(i+win.size/2) }
490
491             if (all(ind.back %in% ind))
492                 ret[i,1] <- sum(rle(diff(x[ind.back]))$length >= seq.size-1)
493
494             if (all(ind.forw %in% ind))
495                 ret[i,2] <- sum(rle(diff(x[ind.forw]))$length >= seq.size-1)
496
497             if (all(ind.both %in% ind))
498                 ret[i,3] <- sum(rle(diff(x[ind.both]))$length >= seq.size-1)
499
500         }
501         apply(ret, 1, max, na.rm=TRUE)
502
503         return(apply(ret, 1, max, na.rm=TRUE))
504
505     }
```

```
506
507
508       ### Strict dancing indicator
509       ## cut.size: Maxium permitted jump size
510       ## p.jumps:  share of jumps in the classification window that are permitted to be
          higher than cut.size
511
512       strict.dancing <- function(x, win.size, cut.size, p.jumps)
513       {
514
515           n <- length(x)
516           ret <- matrix(NA, n, 3)
517           ind <- 1:n
518
519           for (i in 1:n)
520           {
521               ind.back <- (i-win.size+1):i
522               ind.forw <- i:(i+win.size-1)
523
524               if (win.size %% 2) {
525                   ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
526               } else {
527                   ind.both <- (i-win.size/2):(i+win.size/2) }
528
529               if (all(ind.back %in% ind))
530                   ret[i,1] <- sum(abs(diff(x[ind.back])) > cut.size)/win.size < p.jumps
531
532               if (all(ind.forw %in% ind))
533                   ret[i,2] <- sum(abs(diff(x[ind.forw])) > cut.size)/win.size < p.jumps
534
535               if (all(ind.both %in% ind))
536                   ret[i,3] <- sum(abs(diff(x[ind.both])) > cut.size)/win.size < p.jumps
537
538           }
539           return(apply(ret, 1, any, na.rm=TRUE))
540
541       }
542
543
544       ### fuzzy walking indicator
545       ## cut.size: Maxium permitted jump size
546       fuzzy.walking <- function(x, win.size, cut.size)
547       {
```

```
548
549           n <- length(x)
550           ret <- matrix(NA, n, 3)
551           ind <- 1:n
552
553           for (i in 1:n)
554           {
555               ind.back <- (i-win.size+1):i
556               ind.forw <- i:(i+win.size-1)
557
558               if (win.size %% 2) {
559                   ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
560               } else {
561                   ind.both <- (i-win.size/2):(i+win.size/2) }
562
563               if (all(ind.back %in% ind))
564                   ret[i,1] <- all(diff(x[ind.back]) %in% 1:cut.size)
565
566               if (all(ind.forw %in% ind))
567                   ret[i,2] <- all(diff(x[ind.forw]) %in% 1:cut.size)
568
569               if (all(ind.both %in% ind))
570                   ret[i,3] <- all(diff(x[ind.both]) %in% 1:cut.size)
571
572           }
573           return(apply(ret, 1, any, na.rm=TRUE))
574
575       }
576
577
578       ### strict jumping indicator
579       ## win.size: absolute width of classification window
580       ## cut.size: minimal jump size in percent or absolute
581       ## nr:       Vector c(x1, x2)
582       ##           x1 = Minimal number of forward jumps
583       ##           x2 = Minimal number of backward jumps
584       strict.jumping <- function(x, win.size, cut.size, nr)
585       {
586
587           n <- length(x)
588           ret <- matrix(NA, n, 3)
589           ind <- 1:n
590
```

```
591         if (cut.size < 1) cut.size <- cut.size * n
592
593         for (i in 1:n)
594         {
595             ind.back <- (i-win.size+1):i
596             ind.forw <- i:(i+win.size-1)
597
598             if (win.size %% 2) {
599                 ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
600             } else {
601                 ind.both <- (i-win.size/2):(i+win.size/2) }
602
603             if (all(ind.back %in% ind))
604                 ret[i,1]    <-    sum(diff(x[ind.back])>=cut.size)    >=    nr[1]    &
    sum(diff(x[ind.back])<=(-1)*cut.size) >= nr[2]
605
606             if (all(ind.forw %in% ind))
607                 ret[i,1]    <-    sum(diff(x[ind.forw])>=cut.size)    >=    nr[1]    &
    sum(diff(x[ind.forw])<=(-1)*cut.size) >= nr[2]
608
609             if (all(ind.both %in% ind))
610                 ret[i,1]    <-    sum(diff(x[ind.both])>=cut.size)    >=    nr[1]    &
    sum(diff(x[ind.both])<=(-1)*cut.size) >= nr[2]
611
612         }
613         return(apply(ret, 1, any, na.rm=TRUE))
614
615     }
616
617
618     ### jump quantiles
619     ## win.size:  width of classification window in percent or absolute
620     ## quantiles: quantiles to be calculated
621     jump.quantiles <- function(x, win.size, p = c(0.1, 0.3, 0.5, 0.7, 0.9))
622     {
623
624         n <- length(x)
625         ret <- array(NA, c(n, length(p), 3))
626         colnames(ret) <- paste("q",p,sep="")
627         ind <- 1:n
628
629         if (win.size < 1) win.size <- floor(win.size * n)
630
```

```
631         for (i in 1:n)
632         {
633             ind.back <- (i-win.size+1):i
634             ind.forw <- i:(i+win.size-1)
635 
636             if (win.size %% 2) {
637                 ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
638             } else {
639                 ind.both <- (i-win.size/2):(i+win.size/2) }
640 
641             if (all(ind.back %in% ind))
642                 ret[i,,1] <- quantile(diff(x[ind.back]), p=p)
643 
644             if (all(ind.forw %in% ind))
645                 ret[i,,2] <- quantile(diff(x[ind.forw]), p=p)
646 
647             if (all(ind.both %in% ind))
648                 ret[i,,3] <- quantile(diff(x[ind.both]), p=p)
649 
650         }
651         return(apply(ret, c(1,2), max, na.rm=TRUE))
652 
653     }
654 
655 
656     ### abs jump quantiles
657     ## win.size:  width of classification window in percent or absolute
658     ## quantiles: quantiles to be calculated p = c(0.1, 0.3, 0.5, 0.7, 0.9)
659     abs.jump.quantiles <- function(x, win.size, p = c(0.1, 0.3, 0.5, 0.7, 0.9))
660     {
661 
662         n <- length(x)
663         ret <- array(NA, c(n, length(p), 3))
664         colnames(ret) <- paste("q",p,sep="")
665         ind <- 1:n
666 
667         if (win.size < 1) win.size <- floor(win.size * n)
668 
669         for (i in 1:n)
670         {
671             ind.back <- (i-win.size+1):i
672             ind.forw <- i:(i+win.size-1)
673 
```

```
674             if (win.size %% 2) {
675                 ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
676             } else {
677                 ind.both <- (i-win.size/2):(i+win.size/2) }
678 
679             if (all(ind.back %in% ind))
680                 ret[i,,1] <- quantile(abs(diff(x[ind.back])), p=p)
681 
682             if (all(ind.forw %in% ind))
683                 ret[i,,2] <- quantile(abs(diff(x[ind.forw])), p=p)
684 
685             if (all(ind.both %in% ind))
686                 ret[i,,3] <- quantile(abs(diff(x[ind.both])), p=p)
687 
688         }
689         return(apply(ret, c(1,2), max, na.rm=TRUE))
690 
691     }
692 
693 
694     ### longest move
695     ## win.size: absolute width of classification window
696     ## pos (logical): if TRUE forward moves are encountered
697     ## pos (logical): if FALSE backward moves are encountered
698     longest.move <- function(x, win.size, pos = TRUE)
699     {
700 
701         n <- length(x)
702         ret <- matrix(NA, n, 3)
703         ind <- 1:n
704 
705         for (i in 1:n)
706         {
707             ind.back <- (i-win.size+1):i
708             ind.forw <- i:(i+win.size-1)
709 
710             if (win.size %% 2) {
711                 ind.both <- (i-(win.size-1)/2):(i+(win.size-1)/2)
712             } else {
713                 ind.both <- (i-win.size/2):(i+win.size/2) }
714 
715             if (all(ind.back %in% ind))
```

```
716                 ret[i,1]                                                    <-
        max(rle(diff(x[ind.back])>0)$length[rle(diff(x[ind.back])>0)$values == pos])
717
718             if (all(ind.forw %in% ind))
719                 ret[i,2]                                                    <-
        max(rle(diff(x[ind.forw])>0)$length[rle(diff(x[ind.forw])>0)$values == pos])
720
721             if (all(ind.both %in% ind))
722                 ret[i,3]                                                    <-
        max(rle(diff(x[ind.both])>0)$length[rle(diff(x[ind.both])>0)$values == pos])
723
724         }
725         return(apply(ret, 1, max, na.rm=TRUE))
726
727     }
```

7.4.3. Predict writing phases

```
1     ### ////////////////////////////////////////////////
2     ### modeling writing phases (SNF)
3     ### :: Script for classification of writing phases in new corpora of writing
      processes
4     ### update: feb-2013
5     ### Author: Beate Sick
6     ### ////////////////////////////////////////////////
7
8     rm(list = ls())
9
10    ### load packages
11    #source("allPackages.R")
12    library(randomForest)
13
14    ### ------------------------------------------------------------
15    ### Load the random forests model
16    ### ------------------------------------------------------------
17    ### Enter the path of the random forests model
18    mod_rf.path <- "model/mod_rf.RData"
19    load(mod_rf.path)
20    ### Model that was trained with all authors: mod.rf[["allAuthors"]]. For the
      confusion matrix, a model for each author was trained without the data of this
      author. Then this model was used to detect the phases of this author.
21    knownAuthors <- names(mod.rf)[!(names(mod.rf) %in% "allAuthors")]
22
23
24
25    ### ------------------------------------------------------------
```

```
### load data with features (path has to be data/feature/)
### ------------------------------------------------------------
data.feauture.path <- "data/feauture/data_with_feature.RData"
load(data.feauture.path)


### ------------------------------------------------------------
### Identify the authors that are unkown to the model
### unknown author: mod.rf["allAuthors"] will be applied
### known author:   mod.rf["author name"] will be applied
### ------------------------------------------------------------
Authors <- levels(dat$author)
unknownAuthors <- Authors[!(Authors %in% knownAuthors)]

### message
cat(paste(rep("\n", 20), collapse = ""),
    "knownAuthors (mit mod.rf[author name] predict):\n", knownAuthors, "\n\n",
    "unknownAuthors (mit mod.rf[\"allAuthors\"] predict):\n", unknownAuthors)




### ------------------------------------------------------------
### predict
### Attention: Phases manually coded by linguists have to be included
###            in the variable "phases" if a comparison between linguists
###            and model is requested
### ------------------------------------------------------------
pred <- by(data = dat, INDICES = dat$author,
                                    FUN = function(x){
  author <- as.character(unique(x$author))
  print(author)
  ## if the authur is known, use mod.rf[[authorname]], otherwise
  ## mod.rf["allAuthors"]
  if(author %in% knownAuthors){
    predVec <- predict(mod.rf[[author]], newdata = x, type = "response")
    model <- c("mod.rf (author)")
  } else {
    predVec <- predict(mod.rf[["allAuthors"]], newdata = x, type = "response")
    model <- c("mod.rf (all Authors)")
  }

  ### if the random forest can not classify a revision, code it as "unclear"
  if(any(is.na(predVec))){
    lev <- levels(predVec)
    predVec <- addNA(predVec)
    levels(predVec) <- c(lev, "unclear")
```

```
73        }
74
75        names(predVec) <- NULL
76        names(x$phases) <- NULL
77        pred <- data.frame(phases = x$phases, phases.pred = predVec)
78
79        # this condition has to be true, a checking procedure is built in
80        if(!(all(is.na(x$phases)) || all(is.na(predVec)))){
81          # x$phases may have more levels because "pred" returns only phases
82          # that were available in the training. But phases can also get lost
83          # because they are extracted from the author's profile
84          pred$phases.pred <- factor(predVec, levels = union(levels(predVec),
85                                                             levels(x$phases)))
86          pred$phases <- factor(x$phases, levels = levels(pred$phases.pred))
87          confMatrix <- confusionMatrix(pred$phases.pred, pred$phases)$table
88          attr(pred, "ConfusionTableAll") <- confMatrix
89        }
90
91        ## save the model that was used for the classification
92        attr(pred, "model") <- model
93        ## so it is clear to which author the text, the revision in process and
94        ## the revision in product belongs to
95
96        attr(pred, "author") <- author
97        pred$text <- as.factor(x$text)
98        pred$position_nr <- x$position_nr
99        pred$revision_nr <- x$revision_nr
100       return(pred)
101     })
102
103
104
105     ### ------------------------------------------------------------
106     ### predict
107     ### save predicted values
108     ### ------------------------------------------------------------
109     pred.save.path <- "predict/data"
110
111     lapply(pred, FUN = function(x){write.table(x,
112                                                file = paste(pred.save.path, "/",
113                                                             attr(x, "author"),
114                                                             ".csv", sep = ""),
115                                                row.names = FALSE, sep = ";")})
116     save(pred, imp.mat,
117          file = paste(pred.save.path, "/predict.RData", sep = ""))
```

7.5. List of presentations at conferences

Fürer, M. (2017, February). *How to sequence writing phases of television journalists*. Paper presented at the 4th Writing Research Across Borders (WRAB) conference, Bogotá: Pontificia Universidad Javeriana.

Gantenbein, T., Perrin, D., Fürer, M., Luciani, M., & Zampa, M. (2016, November). *"voilàààààà wow!" Verbalizing emotions in collaborative newswriting*. Paper presented at the international conference on language and emotion, Madrid: Universidad Nacional de Educación a Distancia.

Fürer, M. (2016, July). *The scalability of writing phases*. Paper presented at the 15th international conference of the Special Interest Group (SIG) writing of the European Association for Research on Learning and Instruction (EARLI), Liverpool: Liverpool Hope University.

Perrin, D., & Fürer, M. (2015, September). *Applied linguistics and multiple framework approaches. The case of investigating language awareness in the newsroom*. Paper presented at the 48th annual meeting of the British Association of Applied Linguistics (BAAL): Breaking theory. New directions ins applied linguistics, Birmingham: University of Aston.

Fürer, M. (2015, September). *Classifying writing phases. How television journalists sequence their text production processes*. Paper presented at the European conference on language and digital communication. AILA-Europe junior researchers meeting in applied linguistics, Winterthur: Zürcher Hochschule für Angewandte Wissenschaften.

Fürer, M. (2015, November). "When I came back, I suddenly realized…". Applying dynamic systems theory on newswriting. In: D. Perrin (Chair), *Combining research frameworks in applied linguistics*. Symposium conducted at the 5th international Applied Linguistics and Professional Practice (ALAPP) conference Milano: Università degli Studi.

Perrin, D., Fürer, M., Gnach, A., & Gantenbein, T. (2014, February). *Institutional learning from a newsroom minority in times of change.* Paper presented at the Future of Journalism Conference, Cardiff: Cardiff University.

Fürer, M. (2014, February). *Scaling and sequencing writing phases*. Paper presented at the Informal Research Group (IRG) conference: Negotiating methodological challenges in linguistic research, Fribourg: Univérsité de Fribourg.

Fürer, M. (2014, February). *Scaling and sequencing writing phases*. Paper presented at the 3rd Writing Research Across Borders (WRAB) conference, Paris: Université Paris Nanterre.

Fürer, M. (2014, August). *Writing phases reconsidered. Scaling and sequencing writing phases*. Paper presented at the 14th international conference of the Special Interest Group (SIG) writing of the European Association for Research on Learning and Instruction (EARLI), Amsterdam: Universiteit van Amsterdam.

Fürer, M. (2014, August). Writing phases revisited. Scaling and sequencing transmodal writing. In: D. Perrin (Chair), *Transwriting the News. AILA Research Network (REN) on media linguistics*. Symposium conducted at the 7th Association of Applied Linguistics (AILA) world congress: One world, many languages, Brisbane: Brisbane exhibition and convention centre.

Perrin, D., Fürer, M., Gantenbein, T., & Gnach, A. (2013, May). *Transdisciplinary Action Research and Applied Linguistics. The case of investigating and improving newswriting.* Paper presented at the Swiss Association for Applied Linguistics (VALS-ASLA) conference: What is the relevance of linguistic research for society? Questioning the notion of "impact", Basel: Universität Basel.

Fürer, M. (2013, July). *Writing phases reconsidered. Contexts, scales and typologies*. Paper presented at the 19th International Congress of Linguists (ICL), Genève: Université de Genève.

Fürer, M., Gantenbein, T., & Perrin, D. (2013, April). *"voilàààààà wow!" Identity and emotions in collaborative newswriting*. Paper presented at the i-Mean Conference: Identity and Language, Bristol: University of the West of England.

Fürer, M. (2013, April). Knowledge transformation I. Collaborating with policy-makers and media managers. In: D. Perrin (Chair), *Tacit knowledge as the missing link. Knowledge transformation in the Idée Suisse project*. Symposium conducted at the Jahrestagung der Schweizerischen Gesellschaft für Kommunikations- und Medienwissenschaft (SGKM): Transdisziplinarität in der Kommunikations- und Medienwissenschaft. Return on Investment oder vergebliche Liebesmüh?, Winterthur: Zürcher Hochschule für Angewandte Wissenschaften.

Fürer, M., Gantenbein, T., Perrin, D., Sick, B., & Wildi, M. (2012, September). *Modeling writing phases. Interdisciplinary method building – an interim report*. Paper presented at the 7. Tage der Schweizer Linguistik der Schweizerischen Sprachwissenschaftlichen Gesellschaft (SSG), Lugano: Università della Svizzera italiana.

Fürer, M., Gantenbein, T., & Perrin, D. (2012, July). *Modeling writing phases. Interdisciplinary method building – an interim report*. Paper presented at the 13th international conference of the Special Interest Group (SIG) writing of the European Association for Research on Learning and Instruction (EARLI), Porto: Universidade do Porto.

Fürer, M. (2012, December). Knowledge transformation I. Collaborating with policy-makers and media managers. In: D. Perrin (Chair), *Tacit knowledge as the missing link. Knowledge transformation in the Idée Suisse project*. Symposium conducted at the 2nd International Applied Linguistics and Professional Practice (ALAPP) conference, Sidney: University of Technology.

Fürer, M. (2011, September). *Modeling writing phases*. Paper presented at the summer training school for writing process research by the research network on learning to write effectively (ERN-LWE) and the European cooperation in science and technology (COST): Keystroke logging and eye tracking, Antwerp: University of Antwerp.

7.6. Bibliography

Abdel Latif, M. M. M. (2013). What do we mean by writing fluency and how can It be validly measured? *Applied Linguistics, 34*(1), 99–105. doi:10.1093/applin/ams073

Agar, M. (2004). We have met the other and we're all nonlinear: Ethnography as a nonlinear dynamic system. *Complexity, 10*(2), 16-24. doi:10.1002/cptx.20054

Agar, M. H. (2010). On the ethnographic part of the mix. A multi-genre tale of the field. *Organizational Research Methods, 13*(2), 286–303. doi:10.1177/1094428109340040

Alamargot, D., & Chanquoy, L. (2011). Through the Models of Writing: Ten Years After and Vision for the Future. In V. W. Berninger (Ed.), *Past, Present, and Future Contributions of Cognitive Writing Research to Cognitive Psychology*. New York: Taylor & Francis/Routledge, Psychology Press.

Alamargot, D., Terrier, P., & Cellier, J.-M. (2007). *Written documents in the workplace*. Amsterdam et al.: Elsevier.

Andersson, B., Dahl, J., Holmqvist, K., Holsanova, J., Johansson, V., Karlsson, H., . . . Wengelin, Å. (2006). Combining keystroke logging with eye-tracking. In L. Van Waes, M. Leijten, & C. Neuwirth (Eds.), *Writing and digital media* (pp. 166–172). Amsterdam et al.: Elsevier.

Bazerman, C. (2003). What is not institutionally visible does not count: The problem of making activity assessable, accountable, and plannable. In C. Bazerman & D. Russel (Eds.), Writing selves/writing societies: Research from activity perspectives Perspectives on writing: WAC Clearinghouse. Retrieved from https://wac.colostate.edu/books/selves_societies/selves_societies.pdf.

Beaufort, A. (1999). *Writing in the real world. Making transition from school to work*. New York: Teachers College.

Bell, A. (1991). *The language of news media*. Oxford: Blackwell.

Berninger, V. W., Nielsen, K. H., Abbott, R. D., Wijsman, E., & Raskind, W. (2008). Writing problems in developmental dyslexia: Under-recognized and under-treated. *Journal of School Psychology, 46*(1), 1-21. doi:http://doi.org/10.1016/j.jsp.2006.11.008

Boyd-Graber, J. (2016). Machine learning shouldn't be a black box. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, & G. Silvello (Eds.), *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR*

2016, Padua, Italy, March 20–23, 2016. Proceedings (pp. XXIII–XXV). Cham: Springer International Publishing.

Bracewell, R. J. (2003). Tasks, ensembles, and activity. Linkages between text production and situation of use in the workplace. *Written Communication, 20*(4), 511–559.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. doi:10.1023/a:1010933404324

Britton, J., Burgess, T., Martin, N., McLeod, A., & Rosen, H. (1975). *The development of writing abilities*. London: Macmillan.

Brumfit, C. J. (1995). Teacher professionalism and research. In G. Cook & B. Seidlhofer (Eds.), *Principle and practice in applied linguistics: Studies in honour of H. G. Widdowson* (pp. 27–42). Oxford: Oxford University Press.

Bussmann, H. (2008). *Lexikon der Sprachwissenschaft* (4 ed.). Stuttgart: Kröner.

Cameron, D., Frazer, E., Rampton, B., & Richardson, K. (1992). *Researching language. Issues of power and method*. London: Routledge.

Chenoweth, N. A., & Hayes, J. R. (2001). Fluency in Writing: Generating Text in L1 and L2. *Written Communication, 18*(1), 80-98. doi:10.1177/0741088301018001004

Chenu, F., Pellegrino, F., Jisa, H., & Fayol, M. (2014). Interword and intraword pause threshold in writing. *Frontiers in Psychology, 5*(182), 1–7. doi:10.3389/fpsyg.2014.00182

Chomsky, N. (1995). *A minimalist program for linguistic theory* (Vol. 28). Cambridge: MIT Press.

Chomsky, N. (2016). *What kind of creatures are we?* New York: Columbia University Press.

Cook, G. (2003). *Applied linguistics*. Oxford: Oxford University Press.

De Saussure, F. (1916). *Cours de linguistique générale*. Paris/Lausanne: Payot.

Dor, D. (2003). On newspaper headlines as relevance optimizers. *Journal of Pragmatics, 35*, 695–721.

Ehrensberger-Dow, M., & Perrin, D. (2015). Applying a newswriting research approach to translation. In M. Ehrensberger-Dow, S. Göpferich, & S. O'Brien (Eds.), *Interdisciplinarity in translation and interpreting process research* (pp. 79–94). Amsterdam: John Benjamins.

Eklundh, K. S., & Kollberg, P. (2003). Emerging discourse structure: computer-assisted episode analysis as a window to global revision in university students' writing. *Journal of Pragmatics, 35*(6), 869-891. doi:10.1016/s0378-2166(02)00123-6

Elbow, P. (1998a). *Writing with power. Techniques for mastering the writing process*. Oxford et al.: Oxford University Press.

Elbow, P. (1998b). *Writing without teachers* (2nd ed.). New York: Oxford University Press.

Flower, L. S., & Hayes, J. R. (1981). A cognitive process theory of writing. *College Composition and Communication, 32*(4), 365–387.

Fuller, D. (1995). *Development of topic-comment algorithms and test structures in written compositions of students in grades one through nine*. Unpublished doctoral dissertation. University of Washington. Seattle.

Gabriel, M. (2015). *Fields of sense: A new realist ontology*. Edinburgh: Edinburgh University Press.

Glopper, K. d., Kruiningen, J. v., & Hemmen, N. (2014). Context in writing process research. An exploratory analysis of context characteristics in writing process research in educational and workplace settings. In D. Knorr, C. Heine, & J. Engberg (Eds.), *Methods in writing process research* (pp. 15-41). Frankfurt am Main: Peter Lang.

Grabowski, J. (1996). Writing and speaking. Common grounds and differences toward a regulation theory of written language production. In C. M. Levy & S. Ransdell (Eds.), *The science of writing. Theories, methods, individual differences, and applications* (pp. 73–92). Mahwah: Erlbaum.

Grésillon, A., & Lebrave, J.-L. (2008). Linguistique et génétique des textes. Un décalogue. *Le Français moderne*(Numéro spécial publié à l'occasion de son 75e anniversaire), 37–49.

Hayes, J. R. (2012). Modeling and remodeling writing. *Written Communication, 29*(3), 369–388. doi:10.1177/0741088312451260

Ho, T. K. (1995). *Random decision forests.* Paper presented at the 3rd international conference on document analysis and recognition, Montreal.

Jakobsen, A. L. (2006). Research methods in translation. Translog. In K. Sullivan & E. Lindgren (Eds.), *Computer keystroke logging and writing. Methods and applications* (pp. 95–105). Amsterdam: Elsevier.

Janssen, D. (2007). Review: Written documents in the workplace. *Journal of Writing Research, 1*(1), 84–87.

Kellogg, R. T. (1996). A model of working memory in writing. In C. M. Levy & S. Ransdell (Eds.), *The science of writing. Theories, methods, individual differences and applications* (pp. 57–72). Mahwah: Erlbaum.

Knobloch, C. (2000). Historisch-systematischer Aufriß der psychologischen Schreibforschung. In H. Günther & O. Ludwig (Eds.), *Schrift und Schriftlichkeit. Ein interdisziplinäres Handbuch internationaler Forschung* (Vol. 2, pp. 983–991). Berlin: De Gruyter.

Kollberg, P. (1997). *S-notation as a tool for analysing the episodic structure of revisions*. Retrieved from Stockholm:

Kollberg, P. (1998). *S-Notation. A computer based method for studying and representing text composition. Licentiate Thesis.* Stockholm.

Larsen-Freeman, D., & Cameron, L. (2008). *Complex systems and applied linguistics* (2 ed.). Oxford: Oxford University Press.

Leijten, M., & Van Waes, L. (2013). Keystroke logging in writing research: Using inputlog to analyze and visualize writing processes. *Written Communication, 30*(3), 358-392. doi:10.1177/0741088313491692

Leijten, M., & Van Waes, L. (n.d.). Inputlog: A research tool for logging and analyzing writing processes. Retrieved from http://www.inputlog.net

Leijten, M., Van Waes, L., Schriver, K. A., & Hayes, J. R. (2014). Writing in the workplace. Constructing documents using multiple digital sources. *Journal of Writing Research, 5*(3), 285–337.

Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news, 2*, 18–22.

Luginbühl, M., & Perrin, D. (Eds.). (2011). *Muster und Variation in den Medien. Medienlinguistische Perspektiven auf Textproduktion und Text*. Bern et al.: Lang.

MacArthur, C. A., Graham, S., & Fitzgerald, J. (Eds.). (2016). *Handbook of Writing Research, Second Edition*. New York: Guilford Publications.

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. New York: Wiley.

Nilsson, M., & Kollberg, P. (1994). *Trace-it 2.0 user's manual*. Stockholm: Royal Institute of Technology.

Patton, M. Q. (1990). *Qualitative evaluation and research methods*. Newbury Park, CA: Sage.

Perrin, D. (1997). *Journalistische Schreibstrategien optimieren. Dissertationsschrift.* Universität Bern, Bern.

Perrin, D. (2003). Progression analysis (PA). Investigating writing strategies at the workplace. *Journal of Pragmatics, 35*(6), 907–921.

Perrin, D. (2006). Progression analysis. An ethnographic, computer-based multi-method approach to investigate natural writing processes. In L. Van Waes, M. Leijten, & C. Neuwirth (Eds.), *Writing and digital media* (pp. 175–181). Amsterdam et al.: Elsevier.

Perrin, D. (2011). Knowledge map. Retrieved from http://www.news-writing.net/knowledgemap

Perrin, D. (2012). Transdisciplinary action research. Bringing together communication and media researchers and practitioners. *Journal of Applied Journalism and Media Studies, 1*(1), 3–23. doi:10.1386/ajms.1.1.3_1

Perrin, D. (2013). *The linguistics of newswriting*. Amsterdam, New York et al.: John Benjamins.

Perrin, D. (2015). *Medienlinguistik* (3., aktualisierte Auflage ed. Vol. 2503): Konstanz : UVK Verlagsgesellschaft.

Perrin, D., Burger, M., Fürer, M., Gnach, A., Schanne, M., & Wyss, V. (2012). Idée Suisse: Language policy and writing practice of public service media journalists. In M. Torrance, D. Alamargot, M. Castello, F. Ganier, O. Kruse, A. Mangen, L. Tolchinsky, & L. Van Waes (Eds.), *Learning to write effectively. Current trends in European research*. Bingley: Emerald.

Perrin, D., Fürer, M., Gantenbein, T., Sick, B., & Wildi, M. (2011). *From walking to jumping. Statistical modeling of writing processes.* Paper presented at the The JACET 50th commemorative international convention. Challenges for tertiary English education. JACET's role in the next 50 years, Fukuoka.

Potthast, M., Köpsel, S., Stein, B., & Hagen, M. (2016). Clickbait Detection. In N. Ferro, F. Crestani, M.-F. Moens, J. Mothe, F. Silvestri, G. M. Di Nunzio, C. Hauff, & G. Silvello (Eds.), *Advances in Information Retrieval: 38th European Conference on IR Research,*

ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings (pp. 810-817). Cham: Springer International Publishing.

Rijlaarsdam, G., & Van den Bergh, H. (2006). Writing process theory. A functional dynamic approach. In C. A. MacArthur, S. Graham, & J. Fitzgerald (Eds.), *Handbook of writing research. New York: Guilford.* (pp. 41–53). New York: : Guilford.

Rohman, D. G. (1965). Pre-Writing the Stage of Discovery in the Writing Process. *College Composition and Communication, 16*(2), 106-112. doi:10.2307/354885

Ruffner, M. (1981). An empirical approach for the assessment of journalistic writing. *Journalism Quarterly*(1), 77–82.

Russell, D. R. (1997). Writing and genre in higher education and workplaces: A review of studies that use cultural-historical activity theory. *Mind, Culture, and Activity, 4*(4), 224-237. doi:10.1207/s15327884mca0404_2

Scardamalia, M., Bereiter, C., & Steinbach, R. (1984). Teachability of reflective processes in written composition. *Cognitive Science*(8), 173–190.

Schrijver, I., Vaerenbergh, L., Leijten, M., & Van Waes, L. (2014). The translator as a writer: Measuring the effect of writing skills on the translation product. In J. Engberg & D. Knorr (Eds.), *Methods in writing process research [Textproduktion in Medium]*. Bern: Peter Lang.

Severinson-Eklundh, K., & Kollberg, P. (1996a). A computer tool and framework for analyzing online revisions. In C. M. Levy & S. Ransdell (Eds.), *The science of writing. Theories, methods, individual differences and applications* (pp. 163–188). Mahwah: Erlbaum.

Severinson-Eklundh, K., & Kollberg, P. (1996b). Computer tools for tracing the writing process. From keystroke records to S-notation. In G. Rijlaarsdam, H. Van den Bergh, & M. Couzijn (Eds.), *Current research in writing. Theories, models and methodology* (pp. 526–541). Amsterdam: Amsterdam University Press.

Severinson-Eklundh, K., & Kollberg, P. (2003). Emerging discourse structure: computer-assisted episode analysis as a window to global revision in university students' writing. In D. Perrin (Ed.), *The pragmatics of writing. [Journal of Pragmatics. Special Issue 35/6]* (pp. 869–891).

Simpson, J. (2011). Introduction: Applied linguistics in the contemporary world. In J. Simpson (Ed.), *The Routledge handbook of applied linguistics* (pp. 1–8). New York: Routledge.

Sleurs, K., Jacobs, G., & Van Waes, L. (2003). Constructing press releases, constructing quotations: A case study. *Journal of Sociolinguistics, 7*(2), 135–275.

Strömqvist, S., Holmqvist, K., Johansson, V., Karlsson, H., & Wengelin, Å. (2006). What key-logging can reveal about writing. In K. Sullivan & E. Lindgren (Eds.), *Computer keystroke logging and writing. Methods and applications* (pp. 45–72). Amsterdam: Elsevier.

Svalberg, A. M.-L. (2007). State of the Art. Language awareness and language learning. *Language Teaching, 40*(4), 287–308.

T. Kellogg, R., P. Whiteford, A., E. Turner, C., Cahill, M., & Mertens, A. (2013). Working Memory in Written Composition: A Progress Report. *Journal of Writing Research, 5*(2), 159-190. doi:10.17239/jowr-2013.05.02.1

Techsmith. (n.d.). Camtasia. Retrieved from https://www.techsmith.com/camtasia.html

Van Waes, L., & Leijten, M. (2005). Logging writing processes with InputLog. In L. Van Waes & M. Leijten (Eds.), *Writing and digital media* (Vol. 17, pp. 158–165). Oxford: Elsevier.

Vandendaele, A., De Cuypere, L., & Van Praet, E. (2015). Beyond "Trimming the Fat": The Sub-editing Stage of Newswriting. *Written Communication, 32*(4), 368-395. doi:10.1177/0741088315599391

Wengelin, A., Torrance, M., Holmqvist, K., Simpson, S., Galbraith, D., Johansson, V., & Johansson, R. (2009). Combined eyetracking and keystroke-logging methods for studying cognitive processes in text production. *Behav Res Methods, 41*(2), 337-351. doi:10.3758/BRM.41.2.337

Widdowson, H. G. (2000). On the limitations of linguistics applied. *Applied Linguistics, 21*(1), 3–35.

Wildi, M. (2007). *Real-time signal extraction: beyond maximum likelihood principles*. Berlin: Springer.

Wrobel, A. (2000). Phasen und Verfahren der Produktion schriftlicher Texte. In G. Antos, K. Brinker, W. Heinemann, & S. F. Sager (Eds.), *Text- und Gesprächslinguistik. Ein internationales Handbuch zeitgenössischer Forschung* (Vol. 1, pp. 458–472). Berlin/New York: De Gruyter.